Concept 2.6
Bausteinbibliothek IEC
Teil: EXPERTS

01/2007

33002239.03



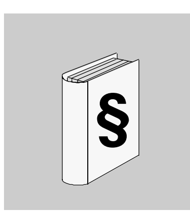


www.telemecanique.com

Inhaltsverzeichnis



Sicherheitshinweise



Wichtige Informationen

HINWEIS

Lesen Sie diese Anweisungen gründlich durch und machen Sie sich mit dem Gerät vertraut, bevor Sie es installieren, in Betrieb nehmen oder warten. Die folgenden Hinweise können an verschiedenen Stellen in dieser Dokumentation enthalten oder auf dem Gerät zu lesen sein. Die Hinweise warnen vor möglichen Gefahren oder machen auf Informationen aufmerksam, die Vorgänge erläutern bzw. vereinfachen.



Erscheint dieses Symbol zusätzlich zu einem Warnaufkleber, bedeutet dies, dass die Gefahr eines elektrischen Schlags besteht und die Nichtbeachtung des Hinweises Verletzungen zur Folge haben kann.



Dies ist ein allgemeines Warnsymbol. Es macht Sie auf mögliche Verletzungsgefahren aufmerksam. Beachten Sie alle unter diesem Symbol aufgeführten Hinweise, um Verletzungen oder Unfälle mit Todesfälle zu vermeiden.

GEFAHR

GEFAHR macht auf eine unmittelbar gefährliche Situation aufmerksam, die bei Nichtbeachtung **unweigerlich** einen schweren oder tödlichen Unfall zur Folge hat.

WARNUNG

WARNUNG macht auf eine möglicherweise gefährliche Situation aufmerksam, die bei Nichtbeachtung **unter Umständen** einen schweren oder tödlichen Unfall oder Beschädigungen an Geräten zur Folge haben kann.

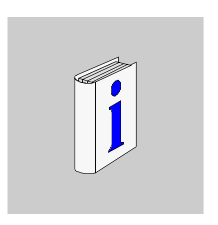
VORSICHT

VORSICHT macht auf eine möglicherweise gefährliche Situation aufmerksam, die bei Nichtbeachtung **unter Umständen** einen schweren oder tödlichen Unfall oder Beschädigungen an Geräten zur Folge hat.

BITTE BEACHTEN

Elektrische Geräte dürfen nur von Fachpersonal installiert, betrieben, gewartet und instand gesetzt werden. Schneider Electric haftet nicht für Schäden, die aufgrund der Verwendung dieses Materials entstehen.

Über dieses Buch



Auf einen Blick

Ziel dieses Dokuments

Diese Dokumentation soll Ihnen bei der Konfiguration der Funktionen und Funktionsbausteine helfen.

Gültigkeits-bereich

Diese Dokumentation gilt für Concept 2.6 unter Microsoft Windows 98, Microsoft Windows 2000, Microsoft Windows XP und Microsoft Windows NT 4.x.

Hinweis: Weitere aktuelle Hinweise finden Sie in der Datei README von Concept.

Weiterführende Dokumentation

Titel	Referenz-Nummer
Concept Installationsanleitung	840 USE 502 02
Concept Benutzerhandbuch	840 USE 503 02
Concept-EFB User Manual	840 USE 505 00
Concept LL984 Bausteinbibliothek	840 USE 506 02

Sie können diese technischen Veröffentlichungen sowie andere technische Informationen von unserer Website herunterladen: *www.telemecanique.com*

Benutzer-kommentar

Ihre Anmerkungen und Hinweise sind uns jederzeit willkommen. Senden Sie sie einfach an unsere E-mail-Adresse: techpub@schneider-electric.com

Allgemeines zur Bausteinbibliothek EXPERTS



Übersicht

Einleitung

Dieser Abschnitt enthält allgemeine Informationen zur Bausteinbibliothek EXPERTS.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

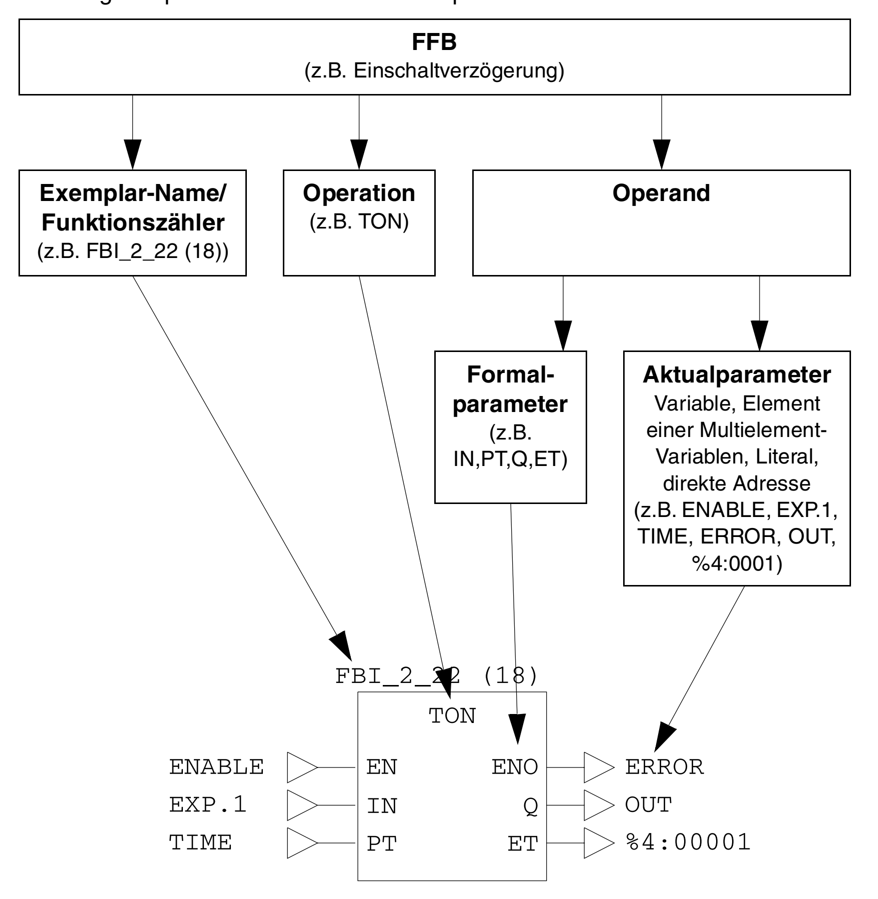
Parametrierung von Funktionen und Funktionsbausteinen

1

Parametrierung von Funktionen und Funktionsbausteinen

Allgemeines

Jeder FFB setzt sich zusammen aus einer Operation, den zur Operation notwendigen Operanden und einem Exemplar-Name/Funktionszähler.



Operation

Die Operation bestimmt, welche Funktionalität durch den FFB ausgeführt werden soll, z.B. Schieberegister, Wandeloperationen.

Operand

Der Operand legt fest, womit die Operation ausgeführt werden soll. Er besteht bei FFBs aus Formalparameter und Aktualparameter.

Formalparameter/ Aktualparameter

Der Formalparameter ist ein Platzhalter für einen Operanden. Bei der Parametrierung wird dem Formalparameter ein aktueller Parameter zugewiesen (Aktualparameter).

Der Aktualparameter kann eine Variable, eine Multielement-Variable, ein Element einer Multielement-Variablen, ein Literal oder eine direkte Adresse sein.

Bedingter/ Unbedingter Aufruf

Jeder FFB verfügt über die Möglichkeit des "bedingten" oder "unbedingten" Aufrufs. Die Bedingung wird über eine Vorverknüpfung des Eingangs EN realisiert.

- EN eingeblendet
  bedingter Aufruf (nur wenn EN = 1, wird der FFB bearbeitet)
- EN ausgeblendet
  unbedingter Aufruf (FFB wird immer bearbeitet)

Hinweis: Wird der EN-Eingang nicht parametriert, muß er ausgeblendet werden. Da nicht parametrierte Eingänge automatisch mit einer "0" belegt werden, würde der FFB sonst nie bearbeitet.

Hinweis: Bei gesperrten Funktionsbausteinen (EN = 0) mit interner Zeitfunktion (z.B. DELAY) läuft die Zeit scheinbar weiter, da sie mit Hilfe der Systemuhr berechnet wird und damit unabhängig vom Programmzyklus und der Freigabe des Bausteins ist.

Aufruf von Funktionen und Funktionsbausteinen in IL und ST

Den Aufruf von Funktionen und Funktionsbausteinen in IL (Anweisungsliste) und ST (Strukturierter Text) entnehmen Sie bitte den enstprechenden Kapiteln des Benutzerhandbuches.

EFB-Beschreibungen



Übersicht

Einleitung

Diese EFB-Beschreibungen sind in alphabetischer Reihenfolge dokumentiert.

Hinweis: Die Anzahl der Eingänge einiger EFBs kann durch vertikale Größenänderung des FFB-Symbols auf max. 32 erhöht werden. Um welche EFBs es sich dabei handelt entnehmen Sie bitte der Beschreibung der einzelnen EFBs.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

ERT_854_10: Datenübertragungs-EFB

2

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein ERT_854_10.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

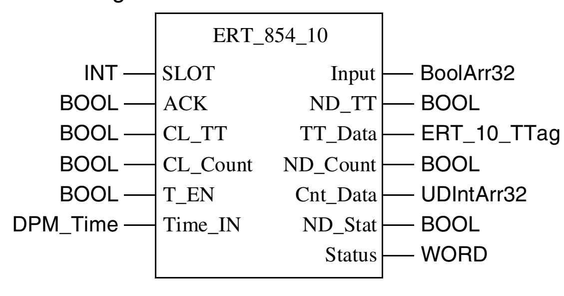
Mit dem ERT_854_10 EFB steht dem Programmierer eine Softwareschnittstelle zur ERT 854 10 Baugruppe zur Verfügung, die einen einfachen Zugriff auf Funktionen wie Zählen, Zeitstempel, Status oder Zeitsynchronisation ermöglicht. Der ERT_854_10 EFB koordiniert den Datenfluß von Multiplex-Daten von der ERT zur SPS über die Ein- und Ausgangsregister. Er sorgt ebenfalls dafür, daß die Zwischenzählwerte in einem internen Speicherbereich abgelegt werden, bis die Daten vollständig sind, sodaß für die Anweisungsliste ein konsistenter Satz aller Zählwerte bereitgestellt wird. Für jeden Datentyp wird ein Merker "Neue Daten" immer dann gesetzt, wenn der Eingangsdatentyp in die entsprechende EFB-Ausgangsstruktur kopiert wurde.

Als zusätzliche Parameter können EN und ENO konfiguriert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
SLOT	INT	Der Slot-Index wird dem ERT-EFB entweder vom QUANTUM- oder DROP-EFB zugewiesen und enthält die konfigurierten Ein- und Ausgangsreferenzen (3x- und 4x-Register)
ACK	BOOL	Ereignis-Bestätigung: das Setzen von ACK signalisiert die Bereitschaft des Anwenders das nächste Ereignis zu empfangen, und löscht den Merker TT_Data. Falls ACK gesetzt bleibt, erfolgt "fortlaufender Betrieb".
CL_TT	BOOL	Löschen des Ereignis-FIFO-Puffer der ERT durch Setzen von CL_TT. Die Speicherung der Ereignisse ist blockiert, bis der CL_TT wieder auf 0 gesetzt ist.
CL_Count	BOOL	Löschen aller ERT-Zähler durch Setzen von CL_Count. Das Zählen wird solange ausgesetzt, bis CL_Count wieder auf 0 gesetzt ist.
T_EN	BOOL	Ermöglicht die Zeitübergabe, z.B. von der ESI, über Time_IN, wenn gesetzt
Time_IN	DPM_Time	Struktur der Eingabezeit, z.B. von der ESI, zur Zeitsynchronisation der ERT (trägt die flankengesteuerte Zeitsynchronisation im Element "Sync")
Input	BOOLArr32	Ausgangsfeld für alle 32 digitalen Eingänge im BOOL-Format (steht auch in Form von Wortreferenzen als 3x-Register 1+2 zur Verfügung)
ND_TT	BOOL	Merker; Neue Daten in TT_Data-Struktur: bleibt gesetzt bis die Bestätigung des Anwenders durch ACK erfolgt
TT_Data	ERT_10_TTag	Ausgangsstruktur der Ereignismeldung mit Zeitmarke. Ein Ereignis wird solange gehalten und NDTT auf 1 gesetzt, bis Anwenderfreigabe mit ACK = 1 erfolgt.
ND_Count	BOOL	Merker; Neue Zählerdaten in Cnt_Data-Struktur: Der Wert 1 liegt nur einen Zyklus an und wird nicht quittiert.
Cnt_Data	UDIntArr32	Ausgangsfeld für 32 Zählerwerte (wird überschrieben, nachdem der EFB einen vollständigen Satz (konfiguriert als: 8, 16, 24, or 32) von zeitkonsistenten Zählwerten erhalten hat.
ND_Stat	BOOL	Merker; Neue Statusdaten im Status-Wort: Der Wert 1 liegt nur einen Zyklus an und wird nicht quittiert.
Status	WORD	Ausgangswort für EFB-/ERT-Status (interne Einzelheiten siehe *Datenfluß, S. 26*)

Interne Zeit-synchronisation

Struktur von DPM_Time für ERT-interne Zeitsynchronisation z.B. durch die ESI:

Element	Elementtyp	Bedeutung
Sync	BOOL	Uhrsynchronisation bei pos. Flanke (zur Stunde oder auf Befehl)
Ms	WORD	Zeit in Millisekunden
Min	BYTE	Zeit ungültig / Minuten
Hour	BYTE	Sommerzeit / Stunden
Day	BYTE	Wochentage / Tag im Monat
Mon	BYTE	Monat
Year	BYTE	Jahr

Ereignisstruktur

Die Ereignisstruktur des ERT_10_TTag mit 5Byte-Zeitmarken (weitere Informationen finden Sie in *Datenfluß, S. 26*):

Element	Elementtyp	Bedeutung
User	BYTE	Vollständige Zeit / Anwendernummer [Modulnummer]
Input	BYTE	Typ der Ereignismenge / Nr. des ersten Eingangs
In	BYTE	Ereignisdaten: 1, 2 oder 8 verwaltete Stellen
Ms	WORD	Zeit in Millisekunden
Min	BYTE	Zeit ungültig / Minuten
Hour	BYTE	Sommerzeit / Stunden
Day	BYTE	Wochentag / Tag des Monats

Funktionsweise

ERT-Daten-übertragung

Die Anzahl der E/A-Worte, die auf dem lokalen und dem Remote-Baugruppenträger zur Verfügung stehen, ist auf 64 Eingänge und 64 Ausgänge begrenzt. Dadurch ist die Anzahl der einsetzbaren ERT-Module pro lokaler/Remote-Rückwandplatine mit den aktuell gewählten minimalen Anforderungen von 7 Eingangs und 5 Ausgangsworten pro Modul auf 9 begrenzt.

Die Größe der benötigten ERT-Datenübertragungen ist wesentlich größer:
- 32 Zähler = 64 Worte,
- ein Ereignis mit 5-Byte-Zeitmerker = 4 Worte,
- 32 digitale Werte und der ERT-Status = 3 Worte.

Aufgrund dieser widersprüchlichen Größenerfordernisse ist der Einsatz eines speziellen Übertragungs-EFB mit dem Namen ERT_854_10 erforderlich, um die erforderlichen Operationen auf der SPS auszuführen und die ERT-Darstellung der Daten in Multiplex-Form anzupassen. Ein solcher EFB ist für jedes einzelne ERT-Modul erforderlich.

Zur Vereinfachung müssen nur diejenigen EFB-Parameter konfiguriert werden, die tatsächlich benutzt werden. Dies erspart einiges an Konfigurationsaufwand, insbesondere, wenn Zähler- und Ereignis-Eingänge nicht miteinander vermischt werden. Leider kann der Speicherplatz dafür nicht gespart werden, da von Concept die Ausgänge mit unsichtbare Dummy-Variablen belegt werden.)

Grundstruktur des ERT_854_10 Eingangs-Registerblocks, mit sieben 3x-Registern für die Übertragung von der ERT zur SPS

Grundstruktur des Registerblocks

ERT_854_10 Eingangs-Registerblock:

Inhalt	Funktion
Digitale Eingänge 1 16	Digital verarbeitete Eingangsdaten, die zylisch aktualisiert werden (die Eingangsadressierung des Moduls entspricht der von digitalen Standard-Eingangsmodulen, d. h. die Eingänge 1 ... 16 entsprechend den Bits 15 ... 0)
Digitale Eingänge 17 32	
Übertragungsstatus	IN-Übertragungsstatus (TS_IN)
MUX 1	Multiplex-Datenblock für die Übertragung von Blöcken, wie:
MUX 2	1 Ereignis mit 5-Byte-Zeitmerker oder
MUX 3	2 Zählerwerte von den maximal konfigurierten 32 oder
MUX 4	1 Statuswort

Vereinfachte Struktur des ERT_854_10 AusgangsRegisterblocks, mit fünf 4xRegistern für die Übertragung von der SPS zur ERT

ERT_854_10 Ausgangs-Registerblock

Inhalt	Funktion
Übertragungsstatus	OUT-Übertragungsstatus (TS_OUT)
MUX 1	Zeit-Datenblock für die ERT für die Synchronisation der Uhr
MUX 2	
MUX 3	
MUX 4	

Hinweis: Anwenderschnittstelle sind normalerweise die Ein- und Ausgänge des ERT_854_10 EFB, nicht die 3x und 4x Register.

EFB-Konfiguration

Anschluß des EFBs

Der Anschluß des EFB an die Eingangs- und Ausgangsreferenzen (3x- und 4x-Register) erfolgt durch eine grafische Verbindung zur ERT-Slotnummer, auf die gleiche Weise wie bei analogen Modulen. Die aktuell verfügbaren QUANTUM- und DROP-EFBs aus der ANA_IO-Bibliothek werden dafür verwendet: QUANTUM für lokale und DROP für Remote-Rückwandplatinen. Diese EFBs übergeben für jeden spezifizierten Slot einen ganzzahligen Index, welcher auf eine interne Datenstruktur mit den konfigurierten Werten verweist. Die Modulparameter und die ID werden dort abgelegt, zusätzlich zu den Adressen und Länge der zugewiesenen Eingangs- und Ausgangsreferenzen (3x- und 4x-Register).

Eine signifikante Verbesserung der Laufzeit kann durch das Deaktivieren des QUANTUM- oder DROP-EFB nach der ersten Ausführung erreicht werden. Die mittlere Laufzeit des ERT_854_10 EFB in einer CPU x13-0x ist ca. 0.6 ms, minimal 0.4 ms, maximal 1.6 ms. Jeder Quantum- oder DROP-EFB läuft im Mittel ca. 1 ms, min. ca. 0.9 ms, max. ca. 1.3 ms.

Datenfluß

Digitale Eingänge

Für diesen Eingangstyp ist kein Merker für neue Daten vorgesehen. Die digitalen Eingänge in den ersten beiden Eingangsregister-Worten werden bei jedem zweiten Zyklus direkt durch die ERT aktualisiert. Der EFB stellt die verarbeiteten Werte als Bool zur Verfügung, wenn das Ausgangsfeld BoolArr32 entsprechend konfiguriert wurde.

Zähler-Eingänge

Die zyklische Aktualisierung der gezählten Werte dauert signifikant länger als bei anderen Datentypen. Gezählte Werte werden als Datensatz in "Cnt_Data" gespeichert, nachdem eine vollständige Folge (konfiguriert als: 8, 16, 24 oder 32) von zeitkonsistenten gezählten Werten in Multiplex-Form von der ERT übertragen wurden. Der Merker für neue Daten "ND_Count" wird für einen Zyklus gesetzt.

Ereignis-eingänge

Dadurch, daß der Anwender die Bereitschaft, neue Ereignisse zu empfangen, aktiv bestätigen muß, wird die Verwaltung von Merkern um einiges komplexer (ein Handshake-Mechanismus ist erforderlich). Ereignisdaten bleiben solange in der Datenstruktur ERT_10_TTag und der Merker für neue Daten "ND_TT" bleibt gesetzt, bis der Anwender den "ACK"-Eingang setzt und damit ein neues Ereignis anfordert. Der EFB antwortet darauf, indem er "ND_TT" für mindestens einen Zyklus zurücksetzt. Nachdem das neue Ereignis in die ERT_10_TT-Struktur Merkerstruktur übertragen wurde, wird "ND_TT" wieder durch den EFB gesetzt. Um zu verhindern, daß die neuen Ereignisdaten überschrieben werden, muß der Anwender darauf achten, daß er den "ACK"-Eingang grundsätzlich zurücksetzt, nachdem der EFB den Merker "ND_TT" zurückgesetzt hat. Dieser Zustand kann dann stabil gehalten werden, um dem Anwenderprogramm ausreichend Zeit für die Ereignisverarbeitung zur Verfügung zu stellen. Jedes nachfolgende Ereignis, das durch die ERT protokolliert wird, wird intern im Ereignis-FIFO-Puffer zwischengespeichert.

Neue Ereignisse werden in Intervallen von mindestens 2 Zyklen direkt vom internen Puffer des EFB übertragen, solange der "ACK"-Eingang gesetzt ist (für den speziellen fortlaufenden Betriebsmodus), welcher wiederum bewirkt, daß der "ND_TT" nur für einen Zyklus gesetzt bleibt. Auch in diesem speziellen Modus bleibt es Aufgabe des Anwenderprogramms die Ereignisverarbeitung zu beenden, bevor "ND_TT" die Übertragung von anderen neuen Ereignissen in die ERT_10_TT-Struktur signalisiert, da in diesem Fall kein Handshake-Schutz durch "ACK" zur Verfügung gestellt wird.

ERT_10_TTag

ERT_10_TTag Ereignis-Struktur mit 5-Byte-Zeitmarken

Byte	Bits	Funktion
1	D0...D6 = Modul-Nr. 0...127 D7 = CT	Grobzeit: CT = 1 zeigt an, daß diese Zeitmarke die gesamte Zeitangabe inkl. Monat und Jahr in den Bytes 2 + 3 enthält. Die Modul-Nr. kann im Parameter-Bildschirm beliebig festgelegt werden.
2	D0...D5 = Eingangsnr. D6 = P1 D7 = P2	Nr. des ersten Eingangs der Ereignisgruppe: 1...32 Typ der Ereignismeldung (P2, P1): 1...3 siehe *Hinweis 1:, S. 27* [Monatwert bei CT = 1]
3	D0...D7 = Daten der Ereignisgruppe (D7...D0, rechtsbündig)	1, 2 oder 8 verarbeitete Stellen [Jahreswert, falls CT = 1]
4	Zeit in Millisekunden (niedrigwertiges Byte	0 ... 59999 Millisekunden (max. 61100) siehe *Hinweis 2:, S. 28*
5	Zeit in Millisekunden (höherwertiges Byte)	0 ... 59999 Millisekunden (max. 61100) siehe *Hinweis 2:, S. 28* und *Hinweis 3:, S. 28*
6	D0...D5 = Minuten D6 = R D7 = TI	Minuten: 0...59 Zeit ungültig: TI = 1 bedeutet ungültige Zeit / reserviert = 0 siehe *Hinweis 3:, S. 28*
7	D0...D4 = Stunden D5 = R D6 = R D7 = DS	Stunden: 0...23 Sommerzeit: DS = 1 zeigt, daß Sommerzeit eingestellt ist Bei Umschaltung SZ -> WZ hat die Stunde 2A die Kennung SZ, die Stunde 2B die Kennung WZ
8	D0...D4 = DOW D5...D7 = DOM	Wochentag: Mo-So = 1...7 Tag des Monats: 1...31 Die Kennung entspricht MEZ und weicht damit von dem in USA verwendeten Standard So = 1 ab.

Hinweis 1:

Interpretation für Byte 2

D7 D6	Typ der Ereignismeldung	D5...D0	Nr. des ersten Eingangs der Ereignisgruppe
0 1	1polige Meldung	1 ... 32	Eingangs-Pinnummer
1 0	2polige Meldung	1, 3, 5, ...31	Erster Eingang der Gruppe
1 1	8polige Meldung	1, 9, 17, 25	Erster Eingang der Gruppe

Hinweis 2: Der Wert für die Millisekunden kann bei Schaltsekunden max. 61100 Millisekunden betragen (61000 plus einer Toleranz von 100 Millisekunden)

Hinweis 3: Bei Zeitmerkern, die eine ungültige Zeit enthalten (TI = 1), wird die Zeit in Millisekunden auf FFFF HEX gesetzt. Minuten, Stunden und DOW/DOM-Werte sind ungültig (d. h. undefiniert).

Grobzeit Ausgabe

Wurde die "Grobzeit Ausgabe" während der ERT-Konfiguration aktiviert, so wird die Übertragung der vollständigen Zeit (mit Monat/Jahr) unter den folgenden Bedingungen ausgeführt: beim Monatswechsel, nach dem Neustart des Moduls, bei jedem Start/Stop des SPS-Anwenderprogramms, beim Löschen des Ereignis-FIFO-Puffer der ERT, beim Starten/Setzen der Uhr. Die Übertragung dieser Grobzeit Ausgabe ohne Werte der Dateneingänge, wird grundsätzlich durch ein richtiges zeitgestempeltes Ereignis "ausgelöst". Geschieht dies nicht, bleiben die Werte solange in der ERT "stecken", bis ein Ereignis stattfindet. Innerhalb der Zeitmarke einer "Grobzeit Ausgabe" ist das CT-Bit immer gesetzt, das Byte 2 enthält die Information über den Monat, Byte 3 über das Jahr und Byte 4 bis 8 zeigen die selben Zeitmarken-Werte des auslösenden Ereignisses, dessen Ereignismeldung der Grobzeit Ausgabe unmittelbar folgt.

Status-Eingänge

Der Merker für neue Status-Daten "ND_Stat" wird für einen Zyklus gesetzt. Die Status-Eingänge können nach 2 Abfragezyklen überschrieben werden.

Das Statuswort enthält EFB und ERT Fehlerbits

Aufteilung der Fehlerbits

Interne Struktur des EFB/ERT-Statusworts:

EFB-Fehlerbits					**ERT-Fehlerbits**										
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

ERT-Fehlerbits

D8 ... D0 ERT-Fehlerbits:

Bit	Kurzbezeichnung	Bedeutung
D0	FW	Firmware-Fehler, Selbsttest-Fehler innerhalb EPROM, RAM oder DPM (Schwerer Modulfehler)
D1	FP	Parametrierungsfehler (schwerer interner Fehler)
D2	TE	externer Zeitreferenz-Fehler (Zeit-Basissignal gestört oder nicht vorhanden)
D3	TU	Zeit wurde ungültig
D4	TA	Zeit ist nicht synchronisiert (Freilauf Mode, Dauerlauf ohne Zeit-Fehler Meldung, siehe auch:*Ohne Gangreserve, S. 33*
D5	PF	FIFO-Puffer-Überlauf (Verlust der neuesten Ereignisdaten)
D6	PH	FIFO-Puffer halbvoll
D7	DC	Entflattern aktiv (Verlust einiger Ereignisdaten)
D8	CE	ERT-Kommunikationsfehler (Prozedurfehler oder Time-Out)

Bei der Konfiguration des Parameter Bildschirms kann ein Teil dieser Fehler der Sammelfehlermeldung durch die "F"-Leuchte sowie dem Fehlerbyte des Moduls innerhalb der Statustabelle zugewiesen werden. Alle anderen Fehler sind dann als Warnungen definiert.

D11 ... D9 reserviert

EFB-Fehlerbits

D15 ... D12 EFB-Fehlerbits:

Bin.	Hex	Bedeutung
1000	8 HEX	EFB-Kommunikations-Time-Out
0101	5 HEX	Falscher Slot
0110	6 HEX	Health-Statusbit ist nicht gesetzt (ERT erscheint als nicht vorhanden)
andere Werte		interner Fehler

Online Fehleranzeige

Die folgenden Fehlermeldungen von ERT/EFB werden im Fenster **Online → Ereignisanzeige** von Concept mit Fehlernummer und Erklärung angezeigt.

EFB-Fehlermeldungen:

Meldung	Fehler	Bedeutung
-2710	Benutzerfehler 11	EFB-Kommunikations-Time-Out
-2711	[Benutzerfehler 12]	EFB interner Fehler
-2712	[Benutzerfehler 13]	EFB interner Fehler
-2713	[Benutzerfehler 14]	EFB interner Fehler
-2714	[Benutzerfehler 15]	EFB interner Fehler
-2715	[Benutzerfehler 16]	Falscher Slot
-2716	[Benutzerfehler 17]	Health-Statusbit ist nicht gesetzt (ERT erscheint als nicht vorhanden)
-2717	[Benutzerfehler 18]	EFB interner Fehler

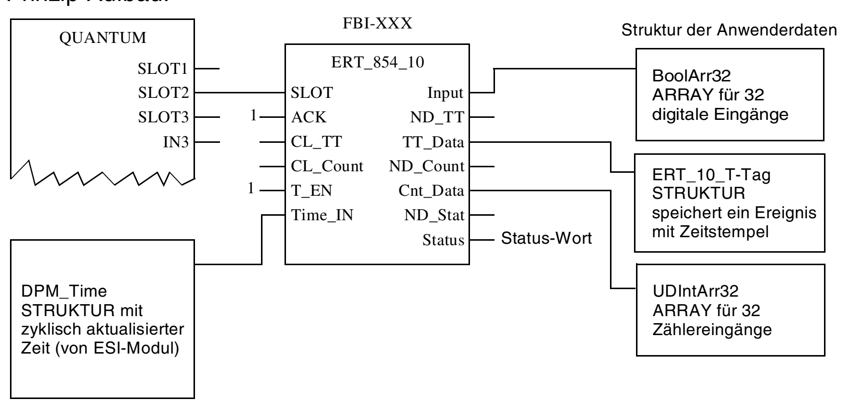
ERT-Fehlermeldungen:

Meldung	Fehler	Bedeutung
-2700	Benutzerfehler 1]	ERT interner Fehler
...	...	...
-2707	[Benutzerfehler 8]	ERT interner Fehler
-2704	[Benutzerfehler 5]	ERT Kommunikations Timeout (z.B. EFB zu lange gesperrt (disabled))

Einfaches Beispiel

Blockschaltbild

Prinzip-Aufbau:



Andere Funktionen

Eingangsmerker

Durch das Setzen des Eingangsmerkers "CL_TT" wird das Löschen des Ereignis-FIFO-Puffers des ERT erzwungen. Das Setzen des Merkers für einen Zyklus ist ausreichend.

Durch das Setzen des Eingangsmerkers "CL_Count" wird das Löschen der ERT-Zähler durch den EFB erzwungen. Das Setzen des Merkers für einen Zyklus ist ausreichend.

Verwendung der DPM_TimeStruktur für die Sychronisation der internen Uhr des ERT

Zeit-synchronisation

Sollte die Zeitsychronisation durch einen Standard-Zeitempfänger nicht zur Verfügung stehen, kann die Zeitinformation alternativ vom 140 ESI 062 01-Kommunikationsmodul übernommen werden. Die ESI stellt die aktualisierte Zeit in einer DPM_TimeStruktur dem EFB direkt über den Parameter "Time_IN" zur Verfügung. Die Datenstruktur kann auch durch das Anwenderprogramm gefüllt und die entsprechenden Bits bedient werden. So kann z.B. die Zeit auch durch die CPU gestellt werden.

Mit Gangreserve

Sobald der "UHR"-Parameter des ERT als "interne Uhr" mit einer Gangreserve ungleich Null (d. h. nicht freilaufend) konfiguriert wurde, muß der EFB für die Synchronisation der internen Uhr der ERT die von der ESI zur Verfügung gestellte Zeit verwenden. Solange bis eine erste Synchronisation stattgefunden hat, sendet der ERT das gesetzte Bit "ungültige Zeit" im "Status"Ausgangswort (Bit 3 TU) zurück.

Die Bedingungen für die erste Synchronisation der internen Uhr des ERT über die Struktur DPM_Time sind:

Der EFBParameter "T_EN" zur Freigabe der Zeiteinstellung muß von 0 auf 1 wechseln.

Die vom ESI zur Verfügung gesetllte Zeit in "Time_IN" muß wie folgt aussehen:
- gültig (d. h. das Bit für die Meldung "Zeit ungültig" im "Min"Wert darf nicht gesetzt sein),
- und die Werte in "Ms" müssen sich kontinuierlich ändern.

Sollten die Zeitdaten später ungültig oder nicht mehr gesetzt werden, wechselt TU erst nach Ablauf der konfigurierten Gangreserve auf 1.

Die Synchronisation/Setzen der internen Uhr der ERT über die DPM_Time-Struktur findet statt, wenn:
- EFB-Parameter "T_EN" zur Freigabe der Zeiteinstellung ist auf 1 gesetzt.
- Die von der ESI zur Verfügung gestellten Zeitdaten in "Time_IN" sind gültig (d. h. das "Zeit ungültig"Bit im "Min"Wert darf nicht gesetzt sein).
- Der Zustand des DPM_Time-Elements "Sync" wechselt von 0 auf 1. Dieser Wechsel wird grundsätzlich durch die 140 ESI 062 01 zu jeder vollen Stunde bewirkt, kann aber ebenso als Ergebnis eines geeigneten Telecontrol-Befehls erfolgen.

Die Genauigkeit der von der ESI auf der ERT synchronisierten Zeit wird beeinflußt sowohl durch die Verzögerung, durch die SPS-Zykluszeit, als auch durch die kumulative Komponente, die die Abweichung der SoftwareUhr der ERT reflektiert (< 360 Millisekunden/Stunde).

Ohne Gangreserve

Wenn der "Uhr"Parameter der ERT als "interne Uhr" im freilaufenden Modus (mit einer Gangreserve von Null) konfiguriert wurde, so startet die interne Uhr mit der Default-Einstellung Stunde 0 am 1/1/1990. In diesem Fall kann die Zeit ebenso durch die Verwendung der DPM_Time-Datenstruktur des 140 ESI 062 01 Moduls gestellt werden, wie oben beschrieben. Da in diesem Modus keine Gangreserve zum "Ablaufen" vorhanden ist, wird die Zeit nie ungültig, und das Bit "Zeit nicht synchronisiert" innerhalb des "Status"-Ausgangswort (Bit 4 TA), das vom EFB zurückgegeben wird, ist immer gesetzt.

Verwendung des ERT > EFB Zeitdaten-Flusses

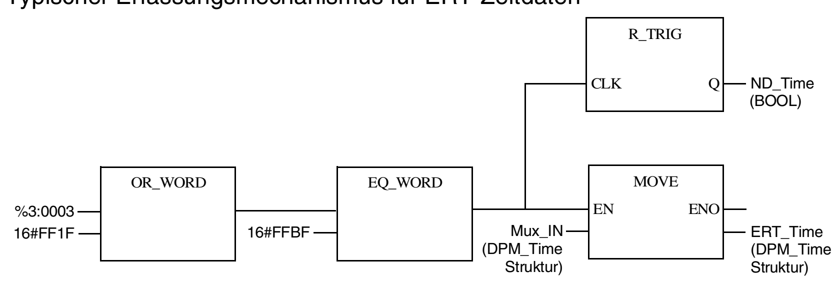
Anwendungs-beispiele:

In diesem Abschnitt wird eine interne Funktion vorgestellt, die durch die ERT für Diagnose und Entwicklung zur Verfügung gestellt wird. Sie umfaßt die zyklische Übertragung der ERT internen Uhrzeit an den entsprechenden EFB in größeren Intervallen. Die Anwendung dieser Zeit, unabhängig davon, ob sie von der freilaufenden internen Uhr stammt oder durch ein externes Referenz-Uhrsignal synchronisiert wurde, kann in der Praxis für Anzeige, Setzen der SPS-Uhr, usw. angewendet werden. Die Zeit erscheint als DPM_Time-Struktur, beginnend bei Wort 4 des IN-Registerblocks des ERT. Das folgende Bild zeigt die zur Selektierung beteiligten Programmelemente.

Inbetriebnahme-Informationen:

Einer ERT_854_10 wurden bei der E/A-Adressierung die IN-Referenzen 30001 ... 30007 zugewiesen. Der IN-Übertragungsstatus (TS_IN) im dritten Wort des Registerblocks wird an einen OR_WORD-Block übergeben. Eine DPM_Time-Struktur wird innerhalb des Variableneditors als Variable Mux_IN im vierten Wort des IN-Registerblocks definiert, und hat damit die Adresse 30004 ... 30007. Diese Variable wird als Eingabe an den MOVE-Block übergeben. Die Ausgabe des MOVE-Blocks ist eine DPM_Time-Struktur, die vom Variableneditor als Variable ERT_Time definiert ist.

Typischer Erfassungsmechanismus für ERT-Zeitdaten



Hinweis: Der ERT_854_10-EFB muß aktiv und fehlerfrei sein.

Erklärung:

Der MOVE-Block überträgt die Zeitdaten, die zyklisch im MUX-Bereich des IN-Registerblocks abgelegt sind, an die DPM_Time-Struktur ERT_Time des Anwenders, sobald der OR_WORT und der EQ_WORD-Block eine Zeitdatenübertragung signalisiert. R_TRIG stellt für einen Zyklus ein Signal in "ND_Time" für die Weiterverarbeitung der Zeitdaten zur Verfügung. Der BOOL"Sync"-Elementwert der ERT_Time sollte bei jeder neuen Übertragung von der ERT zu "ticken" beginnen. Eine neue Übertragung erfolgt jeweils nach maximal 200 SPS-Zyklen.

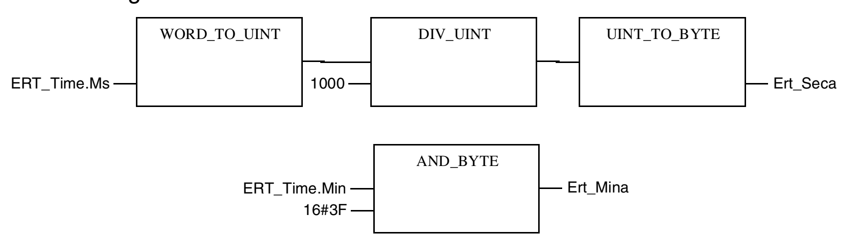
Beispiel 1: Zeitwerte für Anzeige verwenden (oder mit SET_TOD-EFB)

Eine Anzahl von einfachen logischen Operationen ist notwendig, um eine sinnvolle Anzeige der Zeitinformationen der DPM_Time-Struktur zu erhalten. Die gleichen Befehle sind auch für die ERT_10_TTag-Struktur verwendbar. Da im Beispiel 2 das Setzen der SPS-Uhr unter Verwendung des SET_TOD-EFB behandelt wird, werden individuelle Werte direkt in die benötigten Formate umgewandelt.

Hinweis: Der Referenzdateneditor (RDE) kann den "ms"-Wert direkt im Uns-Dec-WORD-Format und den "Min"Wert im Dec-BYTE-Format anzeigen.

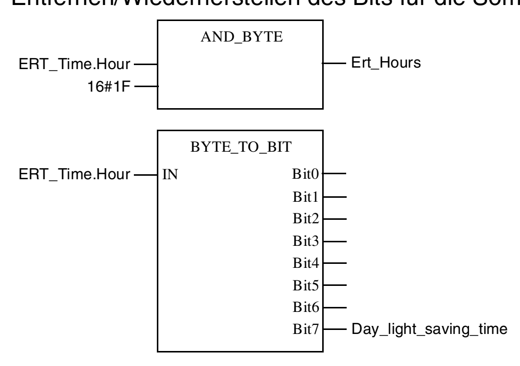
SET_TOD erfordert, daß der WORD-Millisekundenwert "ms" in einen BYTE-Sekundenwert umgewandelt wird. Der BYTE-Minutenwert "Min" enthält das Fehlerbit, das entfernt werden muß (Werte größer 127 sind ungültig).

Umwandlung des WORD-Millisekundenwertes in ein Sekunden-BYTE



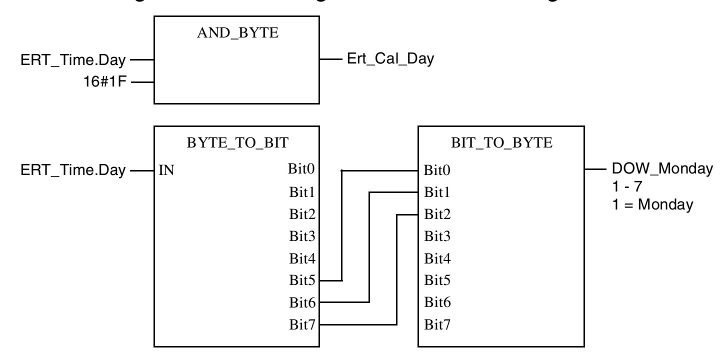
Der BYTE-Wert "Day" enthält sowohl Wochen als auch Kalendertag-Werte. In der DPM_Time-Struktur wird der Wochentag Montag als Wert 1 dargestellt. Im SET_TOD steht der Wochentag-Parameter mit dem Wert 1 für den Sonntag.

Entfernen/Wiederherstellen des Bits für die Sommerzeit vom Wert "Hour"



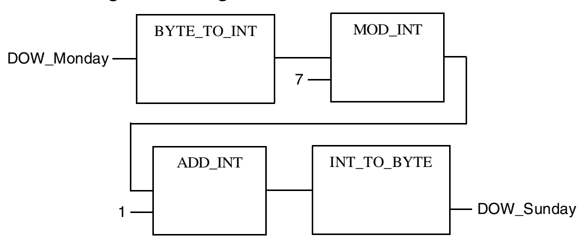
Der BYTE-Wert "Day" enthält sowohl Wochen als auch Kalendertag-Werte. In der DPM_Time-Struktur wird der Wochentag Montag als Wert 1 dargestellt. Im SET_TOD steht der Wochentag-Parameter mit dem Wert 1 für den Sonntag.


Verwendung des Kalendertags und des Wochentags basierend auf Montag



Für die Umwandlung des Wochentags basierend auf dem Wert 1 für Montag in den Wert 1 für Sonntag müssen weitere Schritte ausgeführt werden.

Ausführen von Restwertberechnung (Mod) und Addition zur Umwandlung der Wochentag Wertefolge



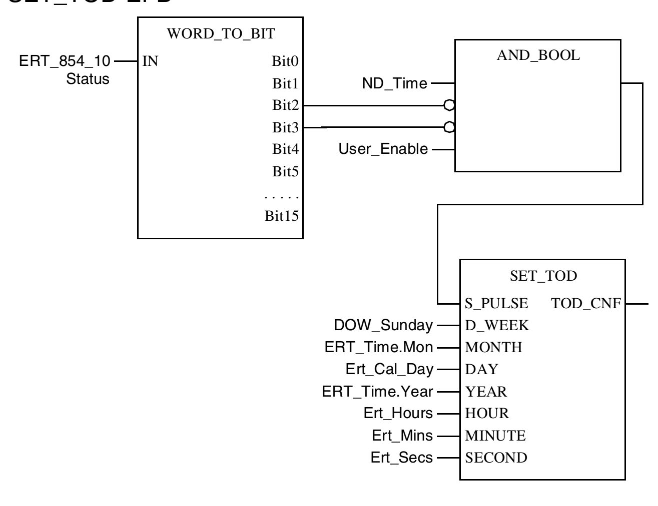
Beispiel 2: Stellen der SPS-Uhr mit dem SET_TOD EFB unter Verwendung von ERTZeitdaten

Alle Parameterwerte, die für den SET_TOD-EFB erforderlich sind, wurden im Beispiel 1 erstellt. Das "ND_Time"-Signal, das für die Übertragung der Zeit in die DPM_Time-Struktur mit dem MOVE-Baustein benötigt wird, wird hier mit einer Anwenderfreigabe kombiniert (z. B. nur einmal pro Stunde), um die SPS-Uhr nur dann zu stellen, wenn neue, fehlerfreie Zeitdaten durch die ERT übertragen wurden. (Die ERT-Fehlerbits werden im Freilauf Modus der internen Uhr nie gesetzt).

Der SET_TOD-EFB ist in der HSBYGruppe der SYSTEM-Bausteinbibliothek enthalten. Wird er verwendet, so ist die Uhr zu aktivieren, indem das TIME OF DAY-Register im Bereich SPECIALS der Konfiguration mit 4x-Adressen belegt wird.

Hinweis: Der "Status"-Parameterwert ist nicht exakt mit dem Zeitdatenstrom synchronisiert, und kann aus diesem Grund nur "tendenziell" den richtigen Wert widerspiegeln.

Vom Anwender freigegebenes Einstellschema für SPS-Uhr unter Verwendung des SET_TOD-EFB



ERT_TIME: Zeitübertragung zur ERT854

3

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein ERT_TIME.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionbaustein liest die DCF-Zeit aus der ESI 062 00 und kann damit die internen Uhren aller ERT-Baugruppen einer TSX Quantum synchronisieren, ohne daß die ERT-Baugruppen selber mit DCF-Empfängern ausgerüstet sein müssen. Der Synchronisationsprozeß findet stündlich statt.

Alle ERT-Baugruppen haben nach den Synchronisationsprozeß die gleiche Uhrzeit. Eine exakte Zeitgleichheit mit der "ESI-Zeit" läßt sich allerdings nicht erreichen. Die Zeitabweichung zur ESI 062 00 ist abhängig von der Position des Bausteins im Anwenderprogramm und von der Programmlaufzeit. Sie ist maximal gleich der Programmlaufzeit, wenn der Baustein direkt am Programmanfang steht.

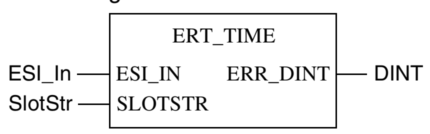
Kern des Funktionsbausteins sind die Parameter ESI_IN und SLOTSTR. ESI_IN ist der Parameter, in dem die ESI 062 "ihre" DCF-Zeit ablegt und in SLOTSTR geben Sie die Steckplätze aller ERT-Baugruppen an, die mit dieser Zeit synchronisiert werden sollen.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbe-schreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
ESI_IN	ESI_In	Datenstruktur, in der die von der ESI empfangene DCF-Zeit abgelegt ist. Geben Sie hier genau den Namen an, den Sie bei der ESI 062 00 verwendet haben.
SLOTSTR	SlotStr	Tragen Sie hier die Steckplätze aller ERT-Baugruppen ein, deren Zeit mit diesem Baustein synchronisiert werden soll.Die Datenstruktur besteht aus 32 Elementen, denen Sie der Reihe nach die Steckplatznummern Ihrer ERT-Baugruppen zuweisen müssen. Die Nummern entsprechen denen in der E/A-Bestückungsliste. Die restlichen Felder müssen "0" sein.Sie können bis zu 14 ERT-Baugruppen eintragen, weitere Einträge werden ignoriert.
ERR_DINT	DINT	Die 32 Bit sind Fehlerbits zu der in SLOTSTR angegebenen ERT-Baugruppen. Jedes Bit entspricht einem Element von SLOTSTR. Die Übertragung ist fehlerfrei verlaufen, wenn alle Bit = 0 sind. Bedeutung der Bits: ● Bit 0 = 1 Fehler bei der Übertragung zur 1. ERT 854 ● Bit 1 = 1 Fehler bei der Übertragung zur 2. ERT 854 ● Bit 3 = 1: ... **Hinweis:** Gezählt werden die Bits von rechts nach links.

EXFR: Istwertfreigabe für Experten

4

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein EXFR.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Die Funktion ist anwendbar für die Expertenbaugruppen 140 ERT 854 00 und 140 ESI 062 00.

Sie schreibt eine "0" auf ein Byte im 3x-Referenzbereich des Experten (Transferstatus), so daß die Übertragung von Daten eines Experten in den Signalspeicher der SPS stattfinden kann.

Nach Ablauf der Übertragung wird die Schreibfreigabe automatisch zurückgesetzt und die Daten sind vor Überschreiben geschützt. Vor dem nächsten Datentransfer muß der EFB erneut aufgerufen werden.

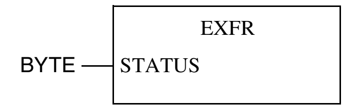
Hinweis: Verwenden Sie diese Funktion in den Textsprachen (IL und ST) nur ein einziges mal, da sonst ein fehlerhafter Code generiet wird.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
STATUS	BYTE	Referenzieren Sie hier nur das Strukturelement "tstat" aus der Datenstruktur des Experten. Beispiel: "xxx.in.tstat" mit xxx als der im Variableneditor vergebene Variablenname für den abgeleiteten Datentyp ESI_In oder ERT_In.

EXRB: Istwerte vom Experten übernehmen

5

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein EXRB.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Die Funktion ist anwendbar für die Expertenbaugruppen 140 ERT 000 00 und 140 ESI 062 00.

Sie kopiert die Istwerte und Statusdaten des Experten aus dem Experten-DPM in den 3x-Referenzbereich des Signalspeichers. Dies erfolgt unmittelbar bei Aufruf des EFBs. Bei der Übertragung von Istwerten müssen Sie zuvor den Zielbereich mit dem EFB "EXFR" freigeben.

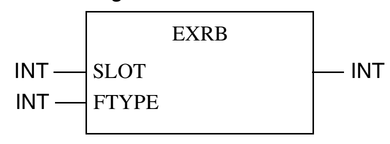
Die Adressen der Register ermittelt die CPU automatisch aus den in der E/A-Bestückungsliste und im Experten-DPM vorliegenden Konfigurationsdaten.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
SLOT	INT	Steckplatznummer der Baugruppe
FTYPE	INT	Feldtyp des zu lesenden Feldes: 1 = Istwertfeld 5 = Statusfeld 1 2, 6, 7 = reserviert für zukünftige Anwendungen. Zur Zeit nicht verwenden!
OUT	INT	Ausführungsstatus des Zugriffs 0 = Feld wurde fehlerfrei gelesen ungleich 0 = Fehler beim Lesen des Feldes (siehe *Laufzeitfehler, S. 47*)

Laufzeitfehler

Fehlermeldung

Fehlermeldungen und ihre Bedeutung:

Fehlernummer	Bedeutung	Abhilfe
-2940	unzulässiger Parameter	EFB-Parametrierung überprüfen
-2941	interner Fehler	wenden Sie sich an die Hotline
-2942	E/A-Operation einer anderen Task aktiv	wenden Sie sich an die Hotline
-2943	Mode = End-Transfer aktiv	EFB später erneut beauftragen
-2945	Zielbereich nicht freigegeben	Zielbereich mit "EXFR" freigeben
-2946	kein Experte bestückt oder falsche Steckplatznummer	EFB-Parametrierung und/oder Eintrag in der E/A-Bestückungsliste überprüfen
-2947	Experte nicht angekoppelt	Hardware-Konfiguration überprüfen, HW-Reset auslösen
-2948	Feld nicht konfiguriert	EFB-Parametrierung überprüfen
-2949	interner Fehler	wenden Sie sich an die Hotline
-2950	kein Experte oder falscher Experte gesteckt	Hardware-Konfiguration überprüfen
-2951	falsche Firmware oder falscher Expertenmodus	Hardware-Konfiguration überprüfen
-2952	Synchronisationsfehler	HW-Reset auslösen
-2953	elementarer Fehler beim Datentransfer	HW-Reset auslösen
-2954	Loadable "ULEX" nicht geladen oder Fehlfunktion von "ULEX"	"ULEX" laden bzw. Hotline anrufen
-2955	Quellpuffer nicht belegt	EFB später erneut beauftragen

EXWB: Sollwerte zum Experten übertragen

6

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein EXWB.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Die Funktion ist anwendbar für die Expertenbaugruppen 140 ERT 000 00 und 140 ESI 062 00.

Sie kopiert die Sollwerte des Experten aus einem 4x-Referenzbereich des Signalspeichers in das Experten-DPM. Dies erfolgt unmittelbar bei Aufruf des EFBs.

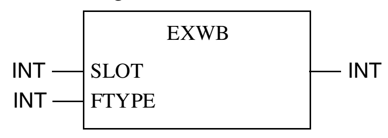
Die Adressen der Register ermittelt die CPU automatisch aus den in der E/A-Bestückungsliste und im Experten-DPM vorliegenden Konfigurationsdaten.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
SLOT	INT	Steckplatznummer der Baugruppe
FTYPE	INT	Feldtyp des zu schreibenden Feldes: 2 = Sollwertfeld 1, 5, 6, 7 = reserviert für zukünftige Anwendungen. Zur Zeit nicht verwenden!
OUT	INT	0 = Feld wurde ohne Fehler geschrieben ungleich 0 = Beim Schreiben des Feldes ist ein Fehler aufgetreten (siehe Laufzeitfehler des Bausteins EXRB (siehe *Laufzeitfehler, S. 47*)).

Laufzeitfehler

Laufzeitfehler

Laufzeitfehler siehe EXRB (siehe *Laufzeitfehler, S. 47*).

MUX_DINTARR_125: Multiplexer für Arrays des Datentyps DIntArr125

7

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein MUX_DINTARR_125.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Mit dieser Funktion können Sie ein einzelnes Element in einem Array adressieren und selektieren.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung des Bausteinparameters MUX_DINTARR_125:

Parameter	Datentyp	Bedeutung
DINTARR	DIntArr125	Array, aus dem ein Element selektiert werden soll
NO	INT	Position im Array, an der das zu selektierende Element seht (Bereich 0 ... 124)
OUT	DINT	selektiertes Element

Beschreibung des Bausteinparameters DIntArr125:

Element	Datentyp	Bedeutung
varname[1]	DINT	1. Element des Arays
...	...	...
varname[125]	DINT	125. Element des Arays

Laufzeitfehler

Fehlermeldung

Es erfolgt eine Fehlermeldung, wenn der zulässige Wertebereich für den Parameter "NO" verletzt wird. Die Fehlernummer wird auch am Ausgang des EFB eingetragen.

MVB_IN: Datenaustausch zwischen CPU und MVB-258A

8

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein MVB_IN.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Dieser Funktionsbaustein realisiert den Datenaustausch zwischen CPU und MVB258A. Die Länge des Datenblocks ist hierbei abhängig von Anzahl und Typ der Variablen. Maximal kann ein Datenblock mit 1024 Worten auf 300 Ports verteilt werden. Der Datenblock wird über die DATASNK-Datenstruktur in die CPU kopiert.

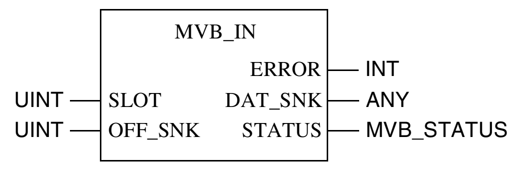
Einschränkungen:
- Zulässige Wortlänge der Sink Ports = 1024
- Maximal können 300 Source und 300 Sink Ports adressiert werden
- Sofern keine Source Ports adressiert werden, können 500 Sink Ports konfiguriert sein
- Der Funktionsbaustein kann mit den Concept-Simulatoren nicht verwendet werden.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbe-schreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
SLOT	UINT	Hier wird der Steckplatz des AS-BMVB-258A im Grundbaugruppenträgers eingetragen. Da die CPU die ersten beiden Steckplätze belegt kann die Kommunikationsbaugruppe nur auf den Steckplätzen 3, 4 oder 5 montiert werden.
OFF_SNK	UINT	Mit diesem Parameter kann ein Offset eingestellt werden. Dieser Wert wird in Bytes angegeben.
ERROR	INT	Der Ausgang ERROR besitzt vier verschiedene Signalzustände:0: 0: keine Fehler 1: ungültige Steckplatzadresse (nur 3 - 5 im Grundträger zulässig) 2: es wurde eine falsche Baugruppe auf diesem Steckplatz plaziert 4: der Datenblock zusammen mit dem Offset ist größer als 1024 oder 0
DATA_SNK	ANY	Dieser Parameter enthält die Eingabe-Datenstruktur (max. 1024 Worte). Der Parameter DATA_SNK hat als Voreinstellung den Datentyp ANY. Sie können somit einen selbst definierten Datentyp verwenden oder den vordefinierten Datentyp MVB_IN, der als Array von 1024 Worten definiert ist verwenden.
STATUS	MVB_STATUS	Der Statusparameter ist ein Array von 32 Worten. Jedes Bit aus dem Array gibt die Gültigkeit von einem Port wieder. Die Bits sind in aufsteigender Reihenfolge der Ports abgelegt. Die Statusbits werden für alle konfigurierten Ports aktuallisiert. Ist das Bit = "0" so ist die Portvariable gültig. Entsprechend ist die Portvariable ungültig sofern das Bit = "1" ist. Alle nicht verwendeten Bits sind auf "1" gesetzt und somit ungültig.

Detailbeschreibung

Laufzeit-optimierung

Um eine bestmögliche Performance hinsichtlich des Laufzeitverhaltens der MVB-Funktionsbausteine zu erzielen, ist die zu übertragende Wortanzahl entsprechend des jeweiligen Anwenderprogramms anzupassen.

Findet keine Optimierung der Arrays statt, werden alle (1024) Worte in jedem Programmzyklus des Anwenderprogramms bearbeitet.

Anpassung der Wortzahl

Wortzahlkonfiguration:

Schritt	Aktion
1	Öffnen Sie unter **c:\Concept\Lib** die Datei EXPERTS.DTY.
2	Ändern Sie die ARRAY-Bereiche entsprechend Ihren Erfordernissen ab.
3	Speichern Sie die Änderungen mit **Datei** → **Speichern** und schließen Sie die Datei.

MVB_INFO: Busdaten über MVB abfragen

9

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein MVB_INFO.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Mit Hilfe des MVB_INFO Funktionsbausteins können Busdaten über den MVB abgefragt werden. So lassen sich Informationen über die Leitung, Konfiguration und Fehlermeldungen anzeigen.

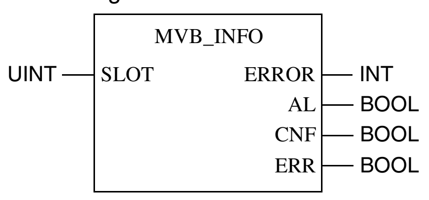
Hinweis: Der Funktionsbaustein kann mit den Concept-Simulatoren nicht verwendet werden.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
SLOT	UINT	Hier wird der Steckplatz des ASB-MVB-258A im Grundbaugruppenträgers eingetragen. Da die CPU die ersten beiden Steckplätze belegt kann die Kommunikationsbaugruppe nur auf den Steckplätzen 3, 4 oder 5 montiert werden.
ERROR	INT	Über den ERROR-Parameter wird die Hardware Konfiguration des MVBs überprüft. Der Parameter kann drei verschiedene Zustände anzeigen: • 0: Fehlerfreie Funktion • 1: Es wurde eine falsche Steckplatzadresse eingetragen • 2: Auf dem parametrierten Steckplatz ist keine Kommunikationsbaugruppe montiert
AL	BOOL	Der AL-Parameter gibt Auskunft über die aktive Leitung. AL=1 bedeutet, daß Leitung_A aktiv. AL= 0 bedeuete, daß Leitung_B aktiv ist.
CNF	BOOL	Über den CNF-Parameter wird der KonfigurationsStatus ausgegeben. Ist der Wert 0 liegt kein Konfigurationsfehler vor und die MVB I/O Task ist aktiv. Wird der Wert 1 ausgegeben so liegt ein Konfigurationsfehler vor.

Detailbeschreibung

Laufzeit-optimierung

Um eine bestmögliche Performance hinsichtlich des Laufzeitverhaltens der MVB-Funktionsbausteine zu erzielen, ist die zu übertragende Wortanzahl entsprechend des jeweiligen Anwenderprogramms anzupassen.

Findet keine Optimierung der Arrays statt, werden alle (1024) Worte in jedem Programmzyklus des Anwenderprogramms bearbeitet.

Anpassung der Wortzahl

Wortzahlkonfiguration

Schritt	Aktion
1	Öffnen Sie unter **c:\Concept\Lib** die Datei EXPERTS.DTY
2	Ändern Sie die ARRAY-Bereiche entsprechend Ihren Erfordernissen ab.
3	Speichern Sie die Änderungen mit **Datei** → **Speichern** und schließen Sie die Datei.

MVB_OUT: Datenaustausch zwischen AS-BMVB-258A und CPU

10

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein MVB_OUT.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein realisiert den Datenaustausch zwischen MVB258A und CPU. Die Länge des Datenblocks ist hierbei abhängig von Anzahl und Typ der Variablen. Maximal kann ein Datenblock mit 1024 Worten auf 300 Ports verteilt werden. Über den DATA_SRC Parameter werden die Datenpakete in die Kommunikationsbaugruppe kopiert.

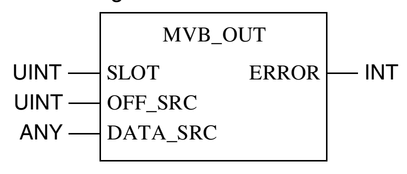
Einschränkungen:
- Die maximale Wortlänge aller Source Ports (inklusive Redundanz Ports wenn konfiguriert) = 1024 Worte.
- Maximal können 300 Source und 300 Sink Ports adressiert werden
- Sofern keine Sink Ports adressiert werden, können 750 Source Ports konfiguriert sein
- Der Funktionsbaustein kann mit den Concept-Simulatoren nicht verwendet werden.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
SLOT	UINT	Hier wird der Steckplatz des AS-BMVB-258A im Grundbaugruppenträgers eingetragen. Da die CPU die ersten beiden Steckplätze belegt kann die Kommunikationsbaugruppe nur auf den Steckplätzen 3, 4 oder 5 montiert werden.
OFF_SRC	UINT	Mit diesem Parameter kann ein Offset eingestellt werden. Dieser Wert wird in Bytes angegeben.
DATA_SRC	ANY	Dieser Parameter enthält den eigentlichen Ausgabedatenblock (max. 1024 Worte).
ERROR	INT	Der Parameterausgang ERROR besitzt vier verschiedene Signalzustände: ● 0: keine Fehler ● 1: ungültige Steckplatzadresse (nur 3-5 im Grundträger zulässig) ● 2: es wurde eine falsche Baugruppe auf diesem Steckplatz plaziert ● 4: der Datenblock zusammen mit dem Offset ist größer als 1024 oder 0

Detailbeschreibung

Laufzeit-optimierung

Um eine bestmögliche Performace hinsichtlich des Laufzeitverhaltens der MVB-Funktionsbausteine zu erziehlen, ist die zu übertragende Wortanzahl entsprechend des jeweiligen Anwenderprogramms anzupassen.

Findet keine Optimierung der Arrays statt, werden alle (1024) Worte in jedem Programmzyklus des Anwenderprogramms bearbeitet.

Anpassung der Wortanzahl

Wortzahlkofiguration

Schritt	Aktion
1	Öffnen Sie unter **c:\Concept\Lib** die Datei EXPERTS.DTY.
2	Ändern Sie die ARRAYBereiche entsprechend Ihren Erfordernissen ab.
3	Speichern Sie die Änderungen mit **Datei** → **Speichern** und schließen Sie die Datei.

MVB_RED: Umschalten der redundanten Source Ports

11

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein MVB_RED.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

MVB_RED ist ein Funktionsbaustein zum Umschalten der redundanten Source Ports. Sind redundante Source Ports definiert, können sie anhand der EFBs aktiv bzw, passiv geschaltet werden.

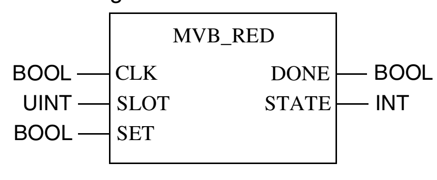
Hinweis: Der Funktionsbaustein kann mit den Concept-Simulatoren nicht verwendet werden.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
CLK	BOOL	Mittels Flankensteuerung wird die Übernahme des SET-Parameters realisiert. Hierbei wird der SET-Parameter nur dann ausgewertet, wenn eine positive Flanke am CLK-Parameter ansteht.
SLOT	UINT	Hier wird der Steckplatz des MVB-258A im Grundbaugruppenträgers eingetragen. Da die CPU die ersten beiden Steckplätze belegt kann die Kommunikationsbaugruppe nur auf den Steckplätzen 3, 4 oder 5 montiert werden.
SET	BOOL	Mit diesem Parameter (Bit) werden die Ports aktiv (1) oder passiv (0) geschaltet.
DONT	BOOL	Dieser Parameter (Antwortbit) gibt über den Status 0 oder 1 den Zustand des redundanten Ports an. D.h. ist der Status = "0" ist der redundante Port passiv geschaltet, ist er ="1" so ist er aktiv.
STATE	INT	Über den Parameter STATE wird der Status bzw. ein Fehlercode ausgegeben. Die einzelnen Meldungen haben folgende Bedeutung: • Statusmeldungen: • 0: Redundanz ist nicht konfiguriert • 1: Umschaltung von passiv nach aktiv • 2: Redundante Ports sind aktiv • 3: Umschaltung von aktiv nach passiv • 4: Redundante Ports sind passiv • Fehlermeldungen: • -1: Es wurde eine falsche Steckplatzadresse eingetragen • -2: Auf dem parametrierten Steckplatz ist keine Kommunikationsbaugruppe montiert

Detailbeschreibung

Laufzeit-optimierung

Um eine bestmögliche Performance hinsichtlich des Laufzeitverhaltens der MVB-Funktionsbausteine zu erzielen, ist die zu übertragende Wortanzahl entsprechend des jeweiligen Anwenderprogramms anzupassen.

Findet keine Optimierung der Arrays statt, werden alle (1024) Worte in jedem Programmzyklus des Anwenderprogramms bearbeitet.

Anpassung der Wortzahl

Wortzahlkonfiguration:

Schritt	Aktion
1	Öffnen Sie unter **c:\Concept\Lib** die Datei EXPERTS.DTY.
2	Ändern Sie die ARRAY-Bereiche entsprechend Ihren Erfordernissen ab.
3	Speichern Sie die Änderungen mit **Datei** → **Speichern** und schließen Sie die Datei.

SIMTSX: TSX-Simulation

12

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein SIMTSX.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Hinweis: Diese Funktion ist nur für interne Verwendung.

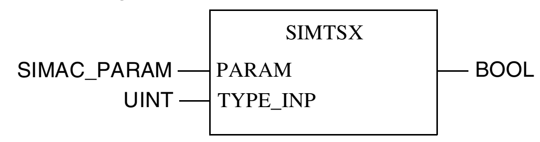
SIMTSX ist eine Funktion, die zusammen mit dem SIMTSX-Softwareprodukt zum Testen von SPS-Programmen verwendet wird.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:



Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
PARAM	SIMAC_PARAM	3 4x-Register
TYPE_INP	UINT	1 = 1x 3 = 3x
STATUS	BOOL	1 = Ausführung OK

ULEXSTAT: Statusmeldungen für Experten

13

Übersicht

Einleitung

Dieses Kapitel beschreibt den Baustein ULEXSTAT.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Kurzbeschreibung

Funktionsbeschreibung

Der Funktionsbaustein ist anwendbar für die Expertenbaugruppen
- 140 NOA 611 00,
- 140 NOA 611 10,
- 140 ERT 854 00 und
- 140 ESI 062 00.

Er liefert detaillierte Informationen über
- Hardwarefehler, die das Loadable "ULEX" erkannt hat
- Softwarefehler, die bei Ausführung des Loadables "ULEX" aufgetreten sind

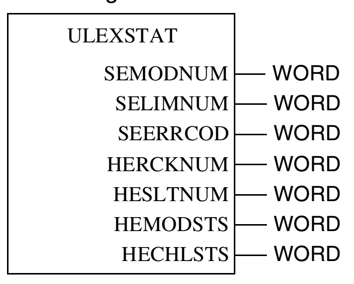
Treten bei mehreren Experten Fehlermeldungen auf, liefert der EFB stets die Statusmeldungen des Experten mit der niedrigsten Steckplatznummer.

Als zusätzliche Parameter können EN und ENO projektiert werden.

Darstellung

Symbol

Darstellung des Bausteins:

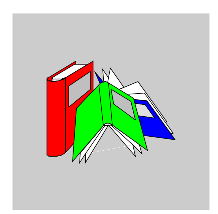


Parameterbeschreibung

Beschreibung der Bausteinparameter:

Parameter	Datentyp	Bedeutung
SEMODNUM	WORD	0: kein Softwarefehler aufgetreten ungleich 0: Softwarefehler. Die angezeigte Nummer dient zur Fehlerlokalisierung. Bitte bei Anfrage in der Hotline bekanntgeben.
SELIMNUM	WORD	siehe oben
SEERRCOD	WORD	siehe oben
HERCKNUM	WORD	0: kein Hardwarefehler erkannt. 1: Hardwarefehler erkannt.
HESLTNUM	WORD	0: kein Hardwarefehler erkannt. 1 .. 16: Steckplatznummer, für den ein Hardware-Fehler erkannt worden ist.
HEMODSTS	WORD	Baugruppenstatus. Identisch mit dem Struktur-Element "USERSTATUS" in der Datenstruktur "EXPSTATUS".
HECHLSTS	WORD	Baugruppenfehlercode. Identisch mit dem Struktur-Element "ERRNO" in der Datenstruktur "EXPSTATUS".

Glossar





A

Abgeleiteter Datentyp Abgeleitete Datentypen sind Datentypen, die von den Elementaren Datentypen und/ oder anderen Abgeleiteten Datentypen abgeleitet wurden. Die Definition der Abgeleiteten Datentypen erfolgt im Datentyp-Editor von Concept.
Es wird zwischen globalen Datentypen und lokalen Datentypen unterschieden.

Ablaufsprache (SFC) Die SFC-Sprachelemente ermöglichen das Unterteilen einer SPS-Programm-Organisationseinheit in eine Anzahl Schritte und Transitionen, die durch gerichtete Verbindungen untereinander verbunden sind. Zu jedem Schritt gehört eine Anzahl Aktionen, und mit jeder Transition ist eine Transitionsbedingung verbunden.

Adressen (Direkte) Adressen sind Speicherbereiche auf der SPS. Diese befinden sich im Signalspeicher und können Ein-/Ausgangs-Baugruppen zugewiesen sein.
Die Anzeige/Eingabe von direkten Adressen ist in den folgenden Formaten möglich:
- Standard-Format (400001)
- Separator-Format (4:00001)
- Kompakt-Format (4:1)
- IEC-Format (QW1)

aktives Fenster Das Fenster, das momentan angewählt ist. Zu einem gegebenen Zeitpunkt kann immer nur ein Fenster aktiv sein. Wenn ein Fenster aktiv wird, ändert sich die Farbe seiner Titelleiste, um es von den anderen Fenstern zu unterscheiden. Nicht angewählte Fenster sind inaktiv.

Aktualparameter Aktuell angeschlossener Eingangs-/Ausgangs-Parameter.

ANL_IN ANL_IN steht für den Datentyp "Analog-Eingang" und wird für die Analogwertverarbeitung verwendet. Dem Datentyp werden die in der E/A-Bestückungsliste festgelegten 3x-Referenzen der konfigurierten Analog-Eingabe-Baugruppe automatisch zugeordnet und dürfen deshalb nur mit Unlocated Variablen belegt werden.

ANL_OUT ANL_OUT steht für den Datentyp "Analog-Ausgang" und wird für die Analogwertverarbeitung verwendet. Dem Datentyp werden die in der E/A-Bestückungsliste festgelegten 4x-Referenzen der konfigurierten Analog-Ausgabe-Baugruppe automatisch zugeordnet und dürfen deshalb nur mit Unlocated Variablen belegt werden.

Anweisung (IL) Anweisungen sind die "Befehle" der Programmiersprache IL. Jede Anweisung beginnt in einer neuen Zeile und wird von einem Operator ggf. mit Modifizierer und, falls erforderlich für die jeweilige Operation, von einem oder mehreren Operanen gefolgt. Falls mehrere Operanden verwendet werden, werden diese durch Kommas getrennt.Vor der Anweisung kann eine Marke stehen, die von einem Doppelpunkt gefolgt wird. Der Kommentar muß, falls vorhanden, das letzte Element der Zeile sein.

Anweisung (ST) Anweisungen sind die "Befehle" der Programmiersprache ST. Anweisungen müssen durch Semikolons abgeschlossen werden. Es können mehrere Anweisungen (durch Semikolons getrennt) in einer Zeile stehen.

Anweisungsliste (IL) IL ist eine Textsprache nach IEC 1131, in der Operationen, wie z. B. bedingte oder unbedingte Aufrufe von Funktionsbausteinen und Funktionen, bedingte oder unbedingte Sprünge usw. durch Anweisungen darsgestellt werden.

Anwendungsfenster Das Fenster, das den Arbeitsbereich, die Menüleiste und die Werkzeugleiste für das Anwendungsprogramm enthält. Der Name des Anwendungsprogramms erscheint in der Titelleiste. Ein Anwendungsfenster kann mehrere Dokumentfenster enthalten.In Concept entspricht das Anwendungsfenster einem Projekt.

ANY In der vorliegenden Version umfaßt "ANY" die Elementaren Datentypen BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME und WORD und davon abgeleitete Datentypen.

ANY_BIT In der vorliegenden Version umfaßt "ANY_BIT" die Datentypen BOOL, BYTE und WORD.

ANY_ELEM In der vorliegenden Version umfaßt "ANY_ELEM" die Datentypen BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME und WORD.

ANY_INT In der vorliegenden Version umfaßt "ANY_INT" die Datentypen DINT, INT, UDINT und UINT.

ANY_NUM In der vorliegenden Version umfaßt "ANY_NUM" die Datentypen DINT, INT, REAL, UDINT und UINT.

ANY_REAL In der vorliegenden Version umfaßt "ANY_REAL" den Datentyp REAL.

Argument Gleichbedeutend mit Aktualparameter.

ASCII-Modus American Standard Code for Information Interchange Der ASCII-Modus wird für die Kommunikation mit verschiedenen Host-Geräten eingesetzt. ASCII arbeitet mit 7 Datenbits.

Atrium Der PC based Controller befindet sich auf einer Standard AT-Platine, und kann innerhalb eines Host-Computers in einem ISA-Bussteckplatz betrieben werden. Die Baugruppe besitzt ein Motherboard (benötigt SA85-Treiber) mit zwei Steckplätzen für PC104-Daughter-Boards. Davon wird ein PC104-Daughter-Board als CPU und das andere zur INTERBUS-Steuerung eingesetzt.

Aufruf Der Vorgang, durch den die Ausführung einer Operation eingeleitet wird.

Ausdruck Ausdrücke bestehen aus Operatoren und Operanden.

Ausgangs-/ Merker-Bits (0x-Referenzen) Ein Ausgangs-/Merker-Bit kann dafür benutzt werden, reale Ausgangsdaten durch eine Ausgabeeinheit des Steuerungssystems zu steuern, oder einen oder mehrere diskrete Ausgänge im Signalspeicher zu definieren.Hinweis: Das x, das nach der ersten Ziffer des Referenztyps steht, repräsentiert einen fünfstelligen Speicherort im Anwender-Datenspeicher, z. B. hat die Referenz 000201 die Bedeutung eines Ausgangs- oder Merkerbits an Adresse 201 des Signalspeichers.

Ausgangs-/ Merkerworte (4x-Referenzen) Ein Ausgangs-/Merkerwort kann für die Speicherung numerischer Daten (binär oder dezimal) im Signalspeicher benutzt werden, oder auch für das Senden der Daten von der CPU an eine Ausgabeeinheit im Steuerungssystem.Hinweis: Das x, das nach der ersten Ziffer des Referenztyps steht, repräsentiert einen fünfstelligen Speicherort im Anwender-Datenspeicher, z. B. hat die Referenz 400201 die Bedeutung eines 16-Bit Ausgangs-/Merkerworts an Adresse 201 des Signalspeichers.

Ausgangs-parameter (Ausgang) Ein Parameter, mit dem das/die Ergebnis(se) der Auswertung eines FFBs zurückgegeben wird/werden.

Auswertung Der Prozeß, durch den ein Wert für eine Funktion oder für die Ausgänge eines Funktionsbausteins während der Programmausführung ermittelt wird.



B

Backup-Datei (Concept-EFB)
Die Backup-Datei ist eine Kopie der letzten Quellcode-Datei. Der Name dieser Backup-Datei ist "backup??.c" (dabei wird angenommen, daß Sie von Ihrer Quellcode-Datei nie mehr als 100 Kopien haben). Die erste Backup-Datei hat den Namen "backup00.c".Falls Sie an der Definitions-Datei Änderungen vorgenommen haben, die beim EFB keine Schnittstellenveränderung hervorrufen, können Sie sich das Erzeugen einer Backup-Datei durch das Editieren Ihrer Quellcode-Datei ersparen (**Objekte** → **Quelle**).Wird eine Backup-Datei angelegt können Sie ihr den Namen Quell-Datei geben.

Basis-16-Literale
Basis-16-Literale dienen zur Angabe von ganzzahligen Werten im Hexadezimal-system. Die Basis muß durch das Präfix 16# gekennzeichnet werden. Die Werte dürfen kein Vorzeichen (+/-) haben. Einzelne Unterstrichzeichen (_) zwischen den Ziffern sind nicht signifikant.

Beispiel
16#F_F oder 16#FF (dezimal 255)
16#E_0 oder 16#E0 (dezimal 224)

Basis-2-Literale
Basis-2-Literale dienen zur Angabe von ganzzahligen Werten im Dualsystem. Die Basis muß durch das Präfix 2# gekennzeichnet werden. Die Werte dürfen kein Vorzeichen (+/-) haben. Einzelne Unterstrichzeichen (_) zwischen den Ziffern sind nicht signifikant.

Beispiel
2#1111_1111 oder 2#11111111 (dezimal 255)
2#1110_0000 oder 2#11100000 (dezimal 224)

Basis-8-Literale
Basis-8-Literale dienen zur Angabe von ganzzahligen Werten im Oktalsystem. Die Basis muß durch das Präfix 8# gekennzeichnet werden. Die Werte dürfen kein Vorzeichen (+/-) haben. Einzelne Unterstrichzeichen (_) zwischen den Ziffern sind nicht signifikant.

Beispiel
8#3_77 oder 8#377 (dezimal 255)
8#34_0 oder 8#340 (dezimal 224)

Bibliothek
Sammlung von Software-Objekten, die für die Wiederverwendung beim Programmieren neuer Projekte vorgesehen sind, oder sogar zum Bauen von neuen Bibliotheken. Beispiele sind die Bibliothek der Elementaren Funktionsbausteintypen.
EFB-Bibliotheken können in Gruppen unterteilt sein.

Binäre Verbindungen
Verbindungen zwischen Ausgängen und Eingängen von FFBs vom Datentyp BOOL.

Bitfolge
Ein Datenelement, das aus einem oder mehreren Bits besteht.

BOOL
BOOL steht für den Datentyp "boolsch". Die Länge der Datenelemente ist 1 Bit (im Speicher abgelegt in 1 Byte). Der Wertebereich für Variablen dieses Datentyps ist 0 (FALSE) und 1 (TRUE).

Bridge
Eine Bridge ist eine Einrichtung, die Netzwerke verbindet. Sie ermöglicht die Kommunikation zwischen Teilnehmern an den zwei Netzwerken. Jedes Netzwerk hat seine eigene Token-Rotationsfolge - das Token wird nicht durch Bridges weitergegeben.

BYTE
BYTE steht für den Datentyp "Bit-Folge 8". Die Eingabe erfolgt als Basis-2-Literal, Basis-8-Literal oder Basis-16-Literal. Die Länge der Datenelemente ist 8 Bit. Ein numerischer Wertebereich kann diesem Datentyp nicht zugeordnet werden.

D

Datentypen
Die Übersicht zeigt die Hierarchie der Datentypen, wie sie bei Ein- und Ausgängen von Funktionen und Funktionsbausteinen verwendet werden. Generische Datentypen sind durch die Vorsilbe "ANY" gekennzeichnet.

- ANY_ELEM
  - ANY_NUM
    ANY_REAL (REAL)
    ANY_INT (DINT, INT, UDINT, UINT)
  - ANY_BIT (BOOL, BYTE, WORD)
  - TIME
- System-Datentypen (IEC-Erweiterungen)
- Abgeleitet (von ’ANY’-Datentypen)

Daten-übertragungs-einstellungen
Einstellungen, die festlegen, wie Informationen von Ihrem Programmiergerät zur SPS übertragen werden.

DCP-E/A-Station Mit einem Distributed Control Prozessor (D908) können Sie ein dezentrales Netzwerk mit übergeordneter SPS einrichten. Beim Einsatz eines D908 mit dezentraler SPS betrachtet die übergeordnete SPS die dezentrale SPS wie eine dezentrale E/A-Station. Die D908 und die dezentrale SPS kommunizieren über den Systembus, wodurch sich eine hohe Leistung bei minimaler Auswirkung auf die Zykluszeit ergibt. Der Datenaustausch zwischen der D908 und der übergeordneten SPS findet mit 1.5 Megabit pro Sekunde über den dezentralen E/A-Bus statt. Eine übergeordnete SPS kann bis zu 31 (Adresse 2-32) D908-Prozessoren unterstützen.

DDE (Dynamic Data Exchange) Die DDE-Schnittstelle ermöglicht einen dynamischen Datenaustausch zwischen zwei Programmen unter Windows. Der Anwender kann die DDE-Schnittstelle im erweiterten Monitor dazu verwenden, seine eigenen Anzeigeanwendungen aufzurufen. Mit dieser Schnittstelle kann der Anwender (d.h. der DDE Client) nicht nur Daten vom erweiterten Monitor (dem DDE Server) lesen, sondern auch Daten an die SPS über den Server schreiben. Der Anwender kann so Daten direkt in der SPS ändern, während er die Ergebnisse überwacht und analysiert.Bei der Verwendung dieser Schnittstelle kann der Anwender sein eigenes "Graphic-Tool", "Face Plate" oder "Tuning Tool" anlegen, und dieses in das System integrieren. Die Tools können in jeder beliebigen Sprache geschrieben werden, z.B. Visual Basic, Visual-C++, die DDE unterstützt. Die Tools werden aufgerufen, wenn der Anwender eine der Schaltflächen im Dialogfeld Erweiterter Monitor betätigt.Concept-Graphic-Tool: Durch die DDE-Verbindung zwischen Concept und Concept Graphic Tool können Signale einer Projektierung als Taktdiagramm dargestellt werden.

Definitions-Datei (Concept-EFB) Die Definitions-Datei beinhaltet allgemeine Beschreibungs-Informationen zum gewählten EFB und seine Formalparameter.

Defragmen-tierung Mit der Defragmentierung werden unbeabsichtigte Lücken (z.B. durch Löschen von unbenutzten Variablen entstanden) im Speicherbereich entfernt.

Deklaration Mechanismus zur Festlegung der Definition eines Sprachelements. Eine Deklaration umfaßt normalerweise die Verbindung eines Bezeichners mit einem Sprachelement und die Zuordnung von Attributen wie Datentypen und Algorithmen.

Derived Function Block (DFB) (Abgeleiteter Funktions-baustein) Ein abgeleiteter Funktionsbaustein repräsentiert den Aufruf eines abgeleiteten Funktionsbausteintyps. Einzelheiten der grafischen Form des Aufrufs finden Sie in der Definition "Funktionsblock (Exemplar)". Im Gegensatz zu Aufrufen von EFB-Typen werden Aufrufe von DFB-Typen mit doppelten vertikalen Linien auf der linken und rechten Seite des rechteckigen Blocksymbols gekennzeichnet.
Der Rumpf eines abgeleiteten Funktionsbausteintyps wird in FBD-Sprache, LD-Sprache, ST-Sprache, IL-Sprache entworfen, allerdings nur in der aktuellen Version des Programmiersystems. Auch abgeleitete Funktionen können in der aktuellen Version noch nicht definiert werden.
Es wird zwischen lokalen und globalen DFBs unterschieden.

Dezentrales Netzwerk (DIO) Eine dezentrale Programmierung im Modbus Plus Netzwerk ermöglicht maximale Leistung beim Datentransfer und keine besonderen Anforderungen an Verknüpfungen. Die Programmierung eines dezentralen Netzes ist einfach. Für die Einrichtung des Netzes braucht keine zusätzliche Kontaktplanlogik erstellt zu werden. Durch entsprechende Einträge im Peer Cop Prozessor werden alle Anforderungen für die Datenübertragung erledigt.

DFB-Code Der DFB-Code ist der ausführbare DFB-Code einer Section. Die Größe des DFB-Codes ist hauptsächlich von der Bausteinanzahl in der Section abhängig.

DFB-Exemplardaten Die DFB-Exemplardaten sind interne Daten der im Programm verwendeten abgeleiteten Funktionsbausteine.

DINT DINT steht für den Datentyp "doppeltlange ganze Zahl (double integer)". Die Eingabe erfolgt als Integer-Literal, Basis-2-Literal, Basis-8-Literal oder Basis-16-Literal. Die Länge der Datenelemente ist 32 Bit. Der Wertebereich für Variablen dieses Datentyps reicht von -2 exp (31) bis 2 exp (31) -1.

Direkte Darstellung Eine Methode zur Variablendarstellung im SPS-Programm, aus der die Zuordnung zum logischen Speicherort - und indirekt zum physikalischen Speicherort - direkt abgeleitet werden kann.

Dokument-fenster Ein Fenster innerhalb eines Anwendungsfensters. Es können mehrere Dokument-fenster gleichzeitig in einem Anwendungsfenster geöffnet sein. Es kann aber immer nur ein Dokumentfenster aktiv sein.Dokumentfenster in Concept sind z. B. Sections, das Meldungs-Fenster, der Referenzdaten-Editor und die SPS-Konfiguration.

DP (PROFIBUS) DP = Dezentrale Peripherie

Dummy Eine Leerdatei, die aus einem Textkopf mit allgemeinen Dateiinformationen, wie z.B. Autor, Erstellungsdatum, EFB-Bezeichnung usw. besteht. Der Anwender muß die Fertigstellung dieser Dummy-Datei durch weitere Einträge vornehmen.

DX Zoom Diese Eigenschaft ermöglich Ihnen das Anschließen an ein Programmierobjekt, um dessen Datenwerte zu beobachten und zu ändern, falls notwendig.



E

E/A-Bestückungs-liste In der E/A-Bestückungsliste werden die E/A- und Experten-Baugruppen der verschiedenen Zentraleinheiten konfiguriert.

EFB-Code
Der EFB-Code ist der ausführbare Code aller verwendeten EFBs. Dazu zählen auch die verwendeten EFBs in DFBs.

Eingangsbits (1x-Referenzen)
Der 1/0-Zustand von Eingangsbits wird durch die Prozeßdaten gesteuert, die von einem Eingabegerät zur CPU gelangen.

> **Hinweis:** Das x, das nach der ersten Ziffer des Referenztyps steht, repräsentiert einen fünfstelligen Speicherort im Anwender-Datenspeicher, z. B. hat die Referenz 100201 die Bedeutung eines Eingangsbits an Adresse 201 des Signalspeichers.

Eingangs-parameter (Eingang)
Übergibt beim Aufruf eines FFBs das zugehörige Argument.

Eingangsworte (3x-Referenzen)
Ein Eingangswort enthält Informationen, die von einer externen Quelle stammen und durch die eine 16-Bit Zahl repräsentiert wird. Ein 3x-Register kann auch 16 aufeinanderfolgende Eingangsbits beinhalten, die in das Register im Binär- oder BCD- (binary coded decimal) Format eingelesen wurden.Hinweis: Das x, das nach der ersten Ziffer des Referenztyps steht, repräsentiert einen fünfstelligen Speicherort im Anwender-Datenspeicher, z. B. hat die Referenz 300201 die Bedeutung eines 16-Bit-Eingangsworts an Adresse 201 des Signalspeichers.

Elementare Funktionen/ Funktionsbau-steine (EFB)
Bezeichnung für Funktionen oder Funktionsbausteine, deren Typdefinitionen nicht in einer der IEC-Sprachen formuliert sind, d.h. deren Rümpfe z. B. nicht mit dem DFB-Editor (Concept-DFB) modifiziert werden können. EFB-Typen werden in "C" programmiert und werden über Bibliotheken in vorkompilierter Form bereitgestellt.

EN / ENO (Freigabe / Fehleranzeige)
Falls der Wert von EN gleich "0" ist, wenn der FFB aufgerufen wird, werden die Algorithmen, die durch den FFB definiert sind, nicht ausgeführt und alle Ausgänge behalten ihren vorherigen Wert. Der Wert von ENO wird in diesem Fall automatisch auf "0" gesetztFalls der Wert von EN gleich "1" ist, wenn der FFB aufgerufen wird, werden die Algorithmen, die durch den FFB definiert sind, ausgeführt. Nach der fehlerfreien Ausführung dieser Algorithmen wird der Wert von ENO automatisch auf "1" gesetzt.Tritt ein Fehler während der Ausführung dieser Algorithmen auf, wird ENO automatisch auf "0" gesetzt.Das Ausgangsverhalten der FFBs ist unabhängig davon, ob die FFBs ohne EN/ENO oder mit EN=1 aufgerufen werden.Ist die Anzeige von EN/ENO eingeschaltet, muß der EN-Eingang unbedingt beschaltet werden. Andernfalls wird der FFB nie ausgeführt.Die Projektierung von EN und ENO wird im Dialogfeld der Bausteineigenschaften ein oder ausgeschaltet. Das Dialogfeld wird über die Menübefehle **Objekte** → **Eigenschaften...** oder über Doppelklick auf den FFB aufgerufen.

Exemplar-Name Ein Bezeichner, der zu einem bestimmten Funktionsbaustein-Exemplar gehört.Der Exemplar-Name dient der eindeutigen Kennzeichnung eines Funktionsbausteines in einer Programm-Organisationseinheit. Der Exemplar-Name wird automatisch erzeugt, kann aber editiert werden. Der Exemplar-Name muß in der gesamten Programm-Organisationseinheit eindeutig sein, dabei wird nicht zwischen Groß-/ Kleinschreibung unterschieden. Existiert der eingegebene Name schon, werden Sie gewarnt und müssen einen anderen Namen wählen. Der Exemplar-Name muß den IEC-Namenskonventionen entsprechen, andernfalls erfolgt eine Fehlermeldung. Der automatisch erzeugte Instanzname hat immer den Aufbau: FBI_n_m

FBI = Funktionsbaustein-Exemplar
n = Nummer der Section (laufende Nummer)
m = Nummer des FFB-Objektes in der Section (laufende Nummer)



F

Fehler Wird bei der Abarbeitung eines FFBs oder eines Schrittes ein Fehler erkannt (z.B. unzulässige Eingangswerte oder ein Zeitfehler), erfolgt eine Fehlermeldung, die Sie sich mit dem Menübefehl **Online** → **Ereignisanzeige...** ansehen können. Bei FFBs wird der ENO-Ausgang auf "0" gesetzt.

Feldvariablen Variablen, denen ein mit Hilfe des Schlüsselworts ARRAY (Feld) ein definierter Abgeleiteter Datentyp zugeordnet ist.Ein Feld ist eine Sammlung von Datenelementen gleichen Datentyps.

FFB (Funktionen/ Funktionsbausteine) Sammelbegriff für EFB (Elementare Funktionen/Funktionsbausteine) und DFB (Abgeleitete (Derived) Funktionsbausteine)

FIR-Filter (Finite Impulse Response Filter) Filter mit endlicher Impulsantwort

Formalparameter Eingangs-/Ausgangsparameter, die innerhalb der Logik eines FFBs verwendet werden und als Ein-/Ausgänge aus dem FFB herausgeführt werden.

Funktion (FUNK) Eine Programm-Organisationseinheit, die bei Ausführung genau ein Datenelement liefert. Eine Funktion hat keine interne Zustandsinformation. Mehrfachaufrufe derselben Funktion mit denselben Eingabeparameterwerten liefern immer dieselben Ausgabewerte.
Einzelheiten der grafischen Form von Funktionsaufrufen finden Sie in der Definition "Funktionsbaustein (Exemplar)". Im Gegensatz zu Aufrufen von Funktionsbausteinen haben Funktionsaufrufe nur einen einzelnen unbenannten Ausgang, da dessen Name der Name der Funktion selbst ist. In FBD wird jeder Aufruf durch eine eindeutige Nummer über dem grafischen Block gekennzeichnet; diese Nummer wird automatisch erzeugt und kann nicht verändert werden.

Funktionsbaustein (Exemplar) (FB) Ein Funktionsbaustein ist eine Programm-Organisationseinheit, die entsprechend der in ihrer Funktionsbausteintypbeschreibung definierten Funktionalität Werte für ihre Ausgänge und interne(n) Variable(n) berechnet, wenn sie als bestimmtes Exemplar aufgerufen wird. Alle Werte der Ausgänge und internen Variablen eines bestimmten Funktionsbaustein-Exemplares bleiben von einem Aufruf des Funktionsbausteins bis zum nächsten bestehen. Mehrfachaufrufe desselben Funktionsbaustein-Exemplars mit denselben Argumenten (Werten von Eingangsparametern) liefern deshalb nicht unbedingt den-/dieselben Ausgabewert(e).
Jedes Funktionsbaustein-Exemplar ist grafisch durch ein rechteckiges Blocksymbol dargestellt. Der Name des Funktionsbausteintyps steht oben in der Mitte innerhalb des Rechtecks. Der Name des Funktionsbaustein-Exemplars steht auch oben, jedoch außerhalb des Rechtecks. Er wird automatisch bei Erstellung einer Instanz generiert, kann aber, je nach Bedarf, vom Anwender abgeändert werden. Eingänge werden auf der linken Seite, Ausgänge auf der rechten Seite des Blocks dargestellt. Die Namen der Formal-Ein-/Ausgangsparameter sind innerhalb des Rechtecks an den entsprechenden Stellen angezeigt.
Die obige Beschreibung der grafischen Darstellung gilt prinzipiell auch für Funktionsaufrufe und für DFB-Aufrufe. Unterschiede sind in den entsprechenden Definitionen beschrieben.

Funktionsbaustein-Sprache (FBD) Eine oder mehrere Sections, die grafisch dargestellte Netzwerke aus Funktionen, Funktionsbausteinen und Verbindungen enthalten.

Funktionsbausteintyp Ein Sprachelement, bestehend aus: 1. der Definition einer Datenstruktur, unterteilt in Eingangs-, Ausgangs-, und interne Variablen; 2. einem Satz von Operationen, die mit den Elementen der Datenstruktur durchgeführt werden, wenn eine Instanz des Funktionbausteintyps aufgerufen wird. Dieser Satz von Operationen kann entweder in einer der IEC-Sprachen (DFB-Typ) oder in "C" (EFB-Typ) formuliert sein.Ein Funktionsbausteintyp kann mehrfach instanziiert (aufgerufen) werden.

Funktionszähler Der Funktionszähler dient der eindeutigen Kennzeichnung einer Funktion in einem Programm oder DFB. Der Funktionszähler ist nicht editierbar und wird automatisch vergeben. Der Funktionszähler hat immer den Aufbau: .n.m

n = Nummer der Section (laufende Nummer)
m = Nummer des FFB-Objektes in der Section (laufende Nummer)



G

Generische Literale Wenn für Sie der Datentyp eines Literal nicht relevant ist, geben Sie einfach den Wert für das Literal an. In diesem Fall weist Concept dem Literal automatisch einen passenden Datentyp zu.

Generischer Datentyp Ein Datentyp, der stellvertretend für mehrere andere Datentypen steht.

Globale Abgeleitete Datentypen Globale Abgeleitete Datentypen sind in jedem Concept-Projekt verfügbar und sind im Verzeichnis DFB direkt unter dem Concept-Verzeichnis abgelegt.

Globale Daten Globale Daten sind Unlocated Variablen.

Globale DFBs Globale DFBs sind in jedem Concept-Projekt verfügbar. Die Ablage der globalen DFBs ist von der Einstellung in der CONCEPT.INI Datei abhängig.

Globale Makros Globale Makros sind in jedem Concept-Projekt verfügbar und sind im Verzeichnis DFB direkt unter dem Concept-Verzeichnis abgelegt.

Gruppen (EFBs) Manche EFB-Bibliotheken (z.B. die IEC-Bibliothek) sind in Gruppen unterteilt. Dies erleichtert besonders in umfangreichen Bibliotheken das Auffinden der EFBs.



H

Hochformat Hochformat bedeutet, daß die Seite bei Betrachtung des gedruckten Textes höher ist als breit.

I

IEC 61131-3
Internationale Norm: Speicherprogrammierbare Steuerungen - Teil 3: Programmiersprachen.

IEC-Format (QW1)
An erster Stelle der Adresse steht ein Bezeichner nach IEC, gefolgt von der fünfstelligen Adresse:
- %0x12345 = %Q12345
- %1x12345 = %I12345
- %3x12345 = %IW12345
- %4x12345 = %QW12345

IEC-Gesamtspeicher
Der IEC-Gesamtspeicher besteht aus dem IEC-Programmspeicher und den Globalen Daten.

IEC-Namenskonvention (Bezeichner)
Ein Bezeichner ist eine Folge von Buchstaben, Ziffern und Unterstrichzeichen, die mit einem Buchstaben oder Unterstrichzeichen beginnen muß (z.B. Name eines Funktionsbaustein-Typs, eines Exemplars, einer Variablen oder einer Section). Buchstaben aus Nationalen Zeichensätzen (z.B: ö,ü, é, õ) können verwendet werden, ausgenommen in Projekt- und DFB-Namen.
Unterstrichzeichen sind in Bezeichnern signifikant; z. B. "A_BCD" und "AB_CD" werden als unterschiedliche Bezeichner interpretiert. Mehrere führende und mehrfache Unterstrichzeichen hintereinander sind nicht zulässig.
Bezeichner dürfen keine Leerstellen enthalten.Groß- bzw. Kleinschreibung ist nicht signifikant; z. B. "ABCD" und "abcd" werden als der gleiche Bezeichner interpretiert.
Bezeichner dürfen keine Schlüsselworte sein.

IEC-Programmspeicher
Der IEC-Programmspeicher besteht aus dem Programm-Code, EFB-Code, den Sectiondaten und den DFB-Exemplardaten.

IIR-Filter
(Infinite Impuls Response Filter) Filter mit unendlicher Impulsantwort

Initialschritt (Anfangsschritt)
Der Startschritt einer Ablaufkette. In jeder Ablaufkette muß ein Initialschritt definiert werden. Mit dem Initialschritt wird die Ablaufkette bei ihrem ersten Aufruf gestartet.

Initialwert
Der einer Variablen beim Programmstart zugeteilte Wert. Die Zuweisung des Wertes erfolgt in Form eines Literals.

Instanziierung
Die Erzeugung eines Exemplars.

Instruktion (LL984)
Beim Programmieren elektrischer Steuerungen hat ein Benutzer die Aufgabe, operational-codierte Anweisungen in Form von Bildobjekten, die in erkennbarer Kontaktform eingeteilt sind, zu implementieren. Die entworfenen Programmobjekte werden, auf Benutzerebene, während des Lade-Prozesses zu computer-verwendbaren OP-Codes konvertiert. Die OP-Codes werden in der CPU entschlüsselt und von den Firmware-Funktionen des Controllers so bearbeitet, daß die gewünschte Steuerung implementiert wird.

INT
INT steht für den Datentyp "ganze Zahl (integer)". Die Eingabe erfolgt als Integer-Literal, Basis-2-Literal, Basis-8-Literal oder Basis-16-Literal. Die Länge der Datenelemente ist 16 Bit. Der Wertebereich für Variablen dieses Datentyps reicht von -2 exp (15) bis 2 exp (15) -1.

Integer-Literale
Integer-Literale dienen der Angabe von ganzzahligen Werten im Dezimalsystem. Die Werte können ein vorangestelltes Vorzeichen (+/-) haben. Einzelne Unterstrichzeichen (_) zwischen den Ziffern sind nicht signifikant.

Beispiel
-12, 0, 123_456, +986

INTERBUS (PCP)
Zur Nutzung des INTERBUS PCP-Kanals und der INTERBUS Prozeßdatenvorverarbeitung (PDV) ist im Concept-Konfigurator der neue E/A-Stationstyp INTERBUS (PCP) eingeführt worden. Diesem E/A-Stationstyp ist die INTERBUS-Anschaltbaugruppe 180-CRP-660-01 fest zugeordnet.
Die 180-CRP-660-01 unterscheidet sich von der 180-CRP-660-00 nur durch einen deutlich größeren E/A-Bereich im Signalspeicher der Steuerung.



Kompakt-Format (4:1)
Die erste Ziffer (die Referenz) wird durch einen Doppelpunkt (:) von der folgenden Adresse getrennt, wobei bei der Adresse die führenden Nullen nicht angegeben werden.

Konstanten
Konstanten sind Unlocated Variablen, denen ein Wert zugewiesen ist, der von der Programmlogik nicht geändert werden kann (schreibgeschützt).

Kontakt
Ein Kontakt ist ein LD-Element, das einen Zustand an die horizontale Verbindung auf seiner rechten Seite übergibt. Dieser Zustand ergibt sich aus der boolschen UND-Verknüpfung des Zustands der horizontalen Verbindung auf seiner linken Seite mit dem Zustand der zugehörigen Variablen/direkten Adresse. Ein Kontakt verändert nicht den Wert der zugehörigen Variablen/direkten Adresse.



L

Ladder Diagram (LD)
Ladder Diagram ist eine grafische Programmiersprache nach IEC1131, die sich optisch an den "Strompfaden" eines Relais-Kontaktplans orientiert.

Ladder Logic 984 (LL)
In den Begriffen Ladder Logic und Ladder Diagram bezieht sich das Wort Ladder (Kontakt) auf Ausführung. Im Gegensatz zu einem Schaltbild, wird ein Kontaktplan von Elektrotechnikern zum Zeichnen eines Stromkreises verwendet (anhand von elektrischen Symbolen), der den Ablauf von Ereignissen zeigen soll und nicht die vorhandenen Drähte, die die Teile miteinander verbinden. Eine gewöhnliche Benutzeroberfläche zum Steuern der Aktionen von Automatisierungsgeräten läßt eine Kontaktplanschnittstelle zu, damit Elektrotechniker zum Implementieren eines Steuerungsprogramms keine Programmiersprache lernen müssen, mit der sie nicht vertraut sind.
Der Aufbau des tatsächlichen Kontaktplanes ermöglicht das Verbinden der elektrischen Elemente auf eine Art und Weise, die einen Steuerungsausgang erzeugt, der von einem logischen Stromfluß durch die verwendeten elektrischen Objekte abhängig ist, die die vorher verlangte Bedingung eines physikalischen Elektrogerätes darstellt.
In einfacher Form ist die Benutzeroberfläche eine von der SPS-Programmieranwendung erarbeitete Video-Anzeige, die ein vertikales und horizontales Raster einrichtet, in das Programmierobjekte eingeordnet werden. Der Plan erhält an der linken Seite des Rasters Strom, und bei Verbindung mit Objekten, die aktiviert sind, fließt der Strom von links nach rechts.

Laufzeitfehler
Fehler, die während der Abarbeitung des Programms auf der SPS, bei SFC-Objekten (z. B. Schritten) oder FFBs auftreten. Dies sind z. B. Wertebereichs-Überläufe bei Zählern oder Zeitfehler bei Schritten.

Literale
Literale dienen dazu, Eingänge von FFBs, Transitionsbedingungen, usw. direkt mit Werten zu versorgen. Diese Werte können von der Programmlogik nicht überschrieben werden (schreibgeschützt). Dabei wird zwischen generischen und typisierte Litralen unterschieden.
Außerdem dienen Literale dazu, einer Konstanten einen Wert oder einer Variablen einen Initialwert zuzuweisen.
Die Eingabe erfolgt als Basis-2-Literal, Basis-8-Literal, Basis-16-Literal, Integer-Literal, Real-Literal oder Real-Literal mit Exponent.

Located Variable Located Variablen wird eine Signalspeicheradresse (Referenzadressen 0x, 1x, 3x,4x) zugeordnet. Der Wert dieser Variablen wird im Signalspeicher gespeichert und kann mit dem Referenzdaten-Editor online geändert werden. Diese Variablen können mit ihrem symbloischen Namen oder mit ihrer Referenzadresse angesprochen werden.

Sämtliche Eingänge und Ausgänge der SPS sind mit dem Signalspeicher verbunden. Der Zugriff des Programms auf Peripherie-Signale, die an der SPS angeschlossen sind, erfolgt nur über Located Variablen. Zugriffe von externer Seite über Modbus- oder Modbus Plus-Schnittstellen der SPS, z.B. von Visualisierungas-systemen, sind ebenfalls über located Variablen möglich.

Lokale Abgeleitete Datentypen Lokale Abgeleitete Datentypen sind nur in einem einzigen Concept-Projekt und dessen lokalen DFBs verfügbar und sind im Verzeichnis DFB unter dem Projekt-Verzeichnis abgelegt.

Lokale DFBs Lokale DFBs sind nur in einem einzigen Concept-Projekt verfügbar und sind im Verzeichnis DFB unter dem Projekt-Verzeichnis abgelegt.

Lokale Makros Lokale Makros sind nur in einem einzigen Concept-Projekt verfügbar und sind im Verzeichnis DFB unter dem Projekt-Verzeichnis abgelegt.

Lokale Verbindung (Local Link) Die lokale Netzwerkverbindung ist das Netzwerk, das den lokalen Teilnehmer mit anderen Teilnehmern entweder direkt oder durch Busverstärker verbindet.

Lokaler Netzwerkteil-nehmer Der lokale Teilnehmer ist der, der gerade projektiert wird.



M

Makro

Makros werden mit Hilfe der Software Concept-DFB erstellt.
Makros dienen zum Duplizieren häufig benutzter Sections und Netzwerke (inklusive deren Logik, deren Variablen und deren Variablendeklaration).
Es wird zwischen lokalen und globalen Makros unterschieden.

Makros haben die folgenden Eigenschaften:
- Makros können nur in den Programmiersprachen FBD und LD erstellt werden
- Makros enthalten nur eine einzige Section
- Makros können eine beliebig komplexe Section enthalten
- Programmtechnisch gesehen unterscheidet sich ein instanziertes Makro, d.h. ein in eine Section eingefügtes Makro, nicht von einer konventionell erstellen Section.
- Auruf von DFBs in einem Makro
- Deklaration von Variablen
- Verwendung von Makro-eigenen Datenstrukturen
- Automatische übernahme der im Makro deklarierten Variablen
- Initialwerte für Variablen
- Mehrfache Instanzierung eines Makros im Gesamtprogramm mit unterschiedlichen Variablen
- Der Sectionname, die Variablennamen und der Datenstrukturname können bis zu 10 verschiedene Austauschmarkierung (@0 bis @9) enthalten.

MMI

Mensch-Maschine-Interface

Multielement-Variablen

Variablen, denen ein mit STRUCT oder ARRAY definierter Abgeleiteter Datentyp zugeordnet ist.
Es wird dabei zwischen Feldvariablen und strukturierten Variablen unterschieden.



N

Netzwerk

Ein Netzwerk ist die Zusammenschaltung von Geräten an einem gemeinsamen Datenweg, die über ein gemeinsames Protokoll miteinander kommunizieren.

Netzwerkteilnehmer

Ein Teilnehmer ist ein Gerät mit einer Adresse (1...64) am Modbus- Plus-Netzwerk.

Node Node ist eine Programmierzelle in einem LL984-Netzwerk. Eine Zelle/Node besteht aus einer 7x11 Matrix, d.h. 7 Reihen zu 11 Elementen.



O

Operand Ein Operand ist ein Literal, eine Variable, ein Funktionsaufruf oder ein Ausdruck.

Operator Ein Operator ist ein Symbol für eine auszuführende arithmetische oder boolsche Operation.



P

Peer-Prozessor Der Peer-Prozessor bearbeitet die Token-Durchläufe und den Datenfluß zwischen dem Modbus-Plus-Netzwerk und der SPS-Anwenderlogik.

Programm Die oberste Programm-Organisationseinheit. Ein Programm wird geschlossen auf eine einzelne SPS geladen.

Programm-Organisations-einheit Eine Funktion, ein Funktionsbaustein, oder ein Programm. Dieser Term kann sich entweder auf einen Typ oder ein Exemplar beziehen.

Programmier-gerät Hardware und Software, die das Programmieren, Projektieren, Testen, Inbetriebnehmen und die Fehlersuche in SPS-Anwendungen sowie in dezentralen Systemanwendungen unterstützt, um Quelldokumentation und -Archivierung zu ermöglichen. Das Programmiergerät kann u. U. auch für die Prozeßvisualisierung benutzt werden.

Programmzyklus Ein Programmzyklus besteht aus dem Einlesen der Eingänge, der Abarbeitung der Programmlogik und der Ausgabe der Ausgänge.

Projekt Allgemeine Bezeichnung für die oberste Ebene einer Software-Baumstruktur, die den übergeordneten Projekt-Namen einer SPS-Anwendung festlegt. Nach der Festlegung des Projekt-Namens können Sie Ihre Systemkonfiguration und Ihr Steuerprogramm unter diesem Namen speichern. Alle Daten, die während der Erstellung der Konfiguration und des Programms entstehen, gehören zu diesem übergeordneten Projekt für diese spezielle Automatisierungs-Aufgabe.
Allgemeine Bezeichnung für den kompletten Satz von Programmier- und Projektierinformation in der Projektdatenbank, die den Quellcode darstellt, der die Automation einer Anlage beschreibt.

Projekt-datenbank Die Datenbank im Programmiergerät, die die Projektierungs-Information für ein Projekt enthält.

Prototyp-Datei (Concept-EFB) Die Prototyp-Datei beinhaltet alle Prototypen der zugeordneten Funktionen. Desweiteren wird, sofern vorhanden, eine Typdefinition der internen Zustandsstruktur angegeben.



Q

Quellcode-Datei (Concept-EFB) Die Quellcode-Datei ist eine gewöhnliche C++ Quelldatei. Nach der Ausführung des Menübefehls **Bibliothek** → **Dateien erzeugen** enthält diese Datei einen EFB-Code-Rahmen, in den Sie einen spezifischen Code für den gewählten EFB eintragen müssen. Hierfür rufen Sie den Menübefehl **Objekte** → **Quelle** auf.

Querformat Querformat bedeutet, daß die Seite bei Betrachtung des gedruckten Textes breiter ist als hoch.



R

REAL

REAL steht für den Datentyp "Gleitkomma-Zahl". Die Eingabe erfolgt als Real-Literal oder als Real-Literal mit Exponent. Die Länge der Datenelemente ist 32 Bit. Der Wertebereich für Variablen dieses Datentyps reicht von +/-3.402823E+38.

> **Hinweis:** Abhängig vom Mathematikprozessortyp der CPU können verschiedene Bereiche innerhalb dieses zulässigen Wertebereichs nicht dargestellt werden. Dies gilt für Werte, die gegen NULL streben und für Werte, die gegen UNENDLICH streben. In diesen Fällen wird im Animationsmodus anstelle eines Zahlenwertes NAN (**N**ot **A** **N**umber) oder INF (**INF**inite (unendlich)) angezeigt.

Real-Literale

Real-Literale dienen zur Angabe von Gleitkomma-Werten im Dezimalsystem. Real-Literale werden durch die Angabe des Dezimalpunktes gekennzeichnet. Die Werte können ein vorangestelltes Vorzeichen (+/-) haben. Einzelne Unterstrichzeichen (_) zwischen den Ziffern sind nicht signifikant.

Beispiel
-12.0, 0.0, +0.456, 3.14159_26

Real-Literale mit Exponent

Real-Literale mit Exponent dienen zur Angabe von Gleitkomma-Werten im Dezimalsystem. Real-Literale mit Exponent werden durch die Angabe des Dezimalpunktes gekennzeichnet. Der Exponent gibt die Zehnerpotenz an, mit der die vorausgehende Zahl zu multiplizieren ist, um den darzustellenden Wert zu erhalten. Die Basis kann ein vorangestelltes negatives Vorzeichen (-) haben. Der Exponent kann ein vorangestelltes positives oder negatives Vorzeichen (+/-) haben. Einzelne Unterstrichzeichen (_) zwischen den Ziffern sind nicht signifikant. (Nur zwischen Ziffern, nicht vor oder hinter dem Dezimalpunkt und nicht vor oder hinter "E", "E+" oder "E-")

Beispiel
-1.34E-12 oder -1.34e-12
1.0E+6 oder 1.0e+6
1.234E6 oder 1.234e6

Redundanzsystem programmieren (Hot Standby)

Ein Redundanzsystem besteht aus zwei identisch konfigurierten SPS-Geräten, die über Redundanzprozessoren miteinander kommunizieren. Im Falle eines Ausfalls der primären SPS übernimmt die sekundäre SPS die Steuerungskontrolle. Unter normalen Bedingungen übernimmt die sekundäre SPS keine Steuerungsfunktion, sondern überprüft die Statusinformationen, um Fehler zu erkennen.

Referenz
Jede direkte Adresse ist eine Referenz, die mit einer Kennung beginnt, die angibt, ob es sich um einen Eingang oder einen Ausgang und ob es sich um ein Bit oder ein Wort handelt. Referenzen, die mit der Kennzahl 6 beginnen, stellen Register im erweiterten Speicher des Signalspeichers dar.
0x Bereich = Ausgangs-/Merkerbits
1x Bereich = Eingangsbits
3x Bereich = Eingangsworte
4x Bereich = Ausgangs-/Merkerworte
6x Bereich = Register im erweiterten Speicher

Hinweis: Das x, das nach der ersten Ziffer jedes Referenztyps steht, repräsentiert einen fünfstelligen Speicherort im Anwender-Datenspeicher, z. B. hat die Referenz 400201 die Bedeutung eines 16-Bit Ausgangs- oder Merkerworts an Adresse 201 des Signalspeichers.

Register im erweiterten Speicher (6x-Referenz)
6x-Referenzen sind Merkerworte im erweiterten Speicher der SPS. Sie können nur bei LL984-Anwenderprogrammen und nur Verwendung einer CPU 213 04 oder CPU 424 02 verwendet werden.

RIO (Remote I/O)
Fern-E/A gibt einen physikalischen Ort der E/A-Punktsteuerungsgeräte an in Bezug auf den sie steuernden Prozessor. Fern-Ein-/Ausgänge sind mit dem Steuerungsgerät über ein verdrahtetes Kommunikationskabel verbunden.

RTU-Modus
Remote Terminal Unit
Der RTU-Modus wird für die Kommunikation zwischen der SPS und einem IBM-kompatiblen Personal Computer verwendet. RTU arbeitet mit 8 Datenbits.



S

SA85-Baugruppe
Die SA85-Baugruppe ist ein Modbus Plus-Adapter für IBM-AT oder kompatible Computer.

Schlüsselworte
Schlüsselworte sind eindeutige Zeichenkombinationen, die als spezielle syntaktische Elemente benutzt werden, wie dies in Anhang B der IEC 1131-3 definiert ist. Alle Schlüsselworte, die in der IEC 1131-3 und damit in Concept benutzt werden, sind im Anhang C der IEC 1131-3 aufgelistet. Diese aufgelisteten Schlüsselworte dürfen für keinen anderen Zweck verwendet werden, z. B. nicht als Variablen-Namen, Section-Namen, Exemplar-Namen, usw.

Schritt
SFC-Sprachelement: Situation, in der das Verhalten eines Programms in Bezug auf seine Eingänge und Ausgänge denjenigen Operationen folgt, die durch die zugehörigen Aktionen des Schrittes definiert sind.

Schrittname
Der Schrittname dient der eindeutigen Kennzeichnung eines Schrittes in einer Programm-Organisationseinheit. Der Schrittname wird automatisch erzeugt, kann aber editiert werden. Der Schrittname muß in der gesamten Programm-Organisationseinheit eindeutig sein, andernfalls erfolgt eine Fehlermeldung.
Der automatisch erzeugte Schrittname hat immer den Aufbau: S_n_m

S = Schritt
n = Nummer der Section (laufende Nummer)
m = Nummer des Schrittes in der Section (laufende Nummer)

Section
Eine Section kann z. B. dazu benutzt werden, die Funktionsweise einer technologischen Einheit, wie eines Motors, zu beschreiben.
Ein Programm oder DFB besteht aus einer oder mehreren Sections. Sections können mit den IEC-Programmiersprachen FBD und SFC programmiert werden. Innerhalb einer Section kann immer nur eine der genannten Programmiersprachen verwendet werden.
Jede Section hat in Concept ihr eigenes Dokumentfenster. Es ist aus Gründen der Übersichtlichkeit allerdings sinnvoll, eine sehr große Section in mehrere kleinere zu unterteilen. Zum Scrollen innerhalb der Section dient die Bildlaufleiste.

Section-Code
Section-Code ist der ausführbare Code einer Section. Die Größe des Section-Codes ist hauptsächlich von der Bausteinanzahl in der Section abhängig.

Sectiondaten
Sectiondaten sind die lokalen Daten in einer Section, wie z.B. Literals, Verbindungen zwischen Bausteinen, nicht verbundene Bausteinein- und ausgänge, interne Zustandsspeicher von EFBs.

Hinweis: Daten, die in DFBs dieser Section vorkommen, sind keine Sectiondaten.

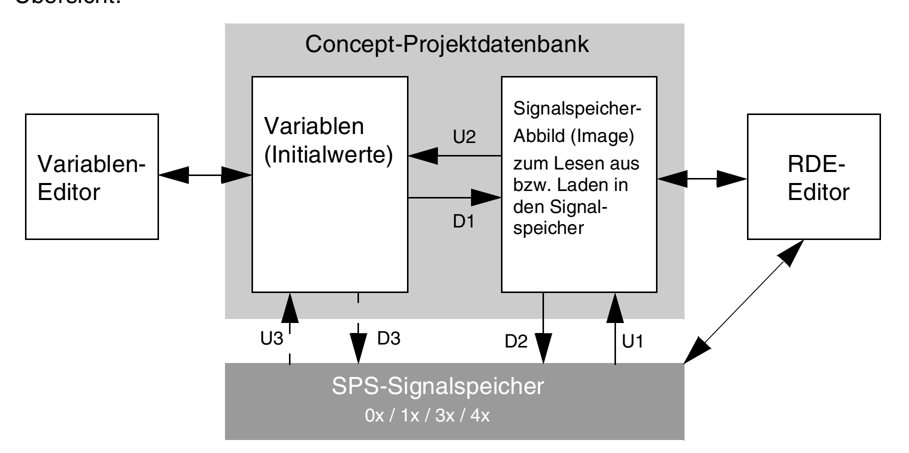
Separator-(Trennzeichen)-Format (4:00001)
Die erste Ziffer (die Referenz) wird durch einen Doppelpunkt (:) von der folgenden fünfstelligen Adresse getrennt.

Serielle Anschlüsse
Bei seriellen Anschlüssen (COM) werden die Informationen bitweise übertragen.

Signalspeicher
Der Signalspeicher ist der Speicherplatz für alle Größen, die über Referenzen (Direkte Darstellung) im Anwenderprogramm angesprochen werden. Zum Beispiel stehen Eingangsbits, Ausgangs-/Merkerbits, Eingangsworte, und Ausgangs-/Merkerworte im Signalspeicher.

Signalspeicherübersicht beim Auslesen und Laden

Übersicht:



Sprachelement

Jedes Grundelement in einer der IEC-Programmiersprachen, z.B. ein Schritt in SFC, ein Funktionsbaustein-Exemplar in FBD oder der Anfangswert einer Variablen.

Sprung

Element der SFC-Sprache. Sprünge werden verwendet, um Bereiche der Ablaufkette zu überspringen.

SPS

Speicherprogrammierbare Steuerung

Spule

Eine Spule ist ein LD-Element, das den Zustand der horizontalen Verbindung auf seiner linken Seite unverändert an die horizontale Verbindung auf seiner rechten Seite übergibt. Dabei wird der Zustand in der zugehörigen Variablen/direkten Adresse gespeichert.

Standard-Format (400001)

Direkt nach der ersten Ziffer (der Referenz) steht die fünfstellige Adresse.

Status-Bits

Zu jedem Teilnehmer mit globaler Eingabe oder spezifischer Ein-/Ausgabe von Peer Cop-Daten gibt es ein Status-Bit. Ist eine definierte Gruppe von Daten erfolgreich innerhalb des eingestellten Timeouts übertragen worden, so wird das entsprechende Status-Bit auf 1 gesetzt. Im anderen Fall wird dieses Bit auf 0 gesetzt und alle zu dieser Gruppe gehörenden Daten (zu 0) gelöscht.

Strukturierte Variablen

Variablen, denen ein mit STRUCT (Struktur) definierter Abgeleiteter Datentyp zugeordnet ist.
Eine Struktur ist eine Sammlung von Datenelementen mit im allgemeinen unterschiedlichen Datentypen (Elementare Datentypen und/oder Abgeleitete Datentypen).

Strukturierter Text (ST)
ST ist eine Textsprache nach IEC 1131, in der Operationen, wie z. B. Aufrufe von Funktionsbausteinen und Funktionen, bedingte Ausführung von Anweisungen, Wiederholung von Anweisungen usw. durch Anweisungen darsgestellt werden.

SY/MAX
In Quantum-Steuergeräten schließt Concept die Bereitstellung auf E/A-Bestückung SY/MAX-E/A-Baugruppen für RIO-Steuerung durch die Quantum-SPS mit ein. Der SY/MAX-Fernbaugruppenträger hat einen Fern-E/A-Adapter in Steckplatz 1, der über ein Modicon S908 R E/A-System kommuniziert. Die SY/MAX-E/A-Baugruppen sind für Sie zur Markierung und Einbeziehung in die E/A-Bestückung der Concept-Konfiguration aufgeführt.

Symbol (Icon)
Grafische Darstellung verschiedener Objekte in Windows, z. B. Laufwerke, Anwendungsprogramme und Dokumentfenster.



T

Teilnehmer-Adresse
Die Teilnehmer-Adresse dient der eindeutigen Kennzeichnung eines Netzwerkteilnehmers im Routing-Pfad. Die Adresse wird direkt am Teilnehmer eingestellt, z.B. durch Drehschalter an der Rückseite der Baugruppen.

Template-Datei (Concept-EFB)
Die Template-Datei ist eine ASCII-Datei mit einer Layout-Information für den Concept FBD-Editor, und den Parametern für die Code-Erzeugung.

TIME
TIME steht für den Datentyp "Zeitdauer". Die Eingabe erfolgt als Zeitdauer-Literal. Die Länge der Datenelemente ist 32 Bit. Der Wertebereich für Variablen dieses Datentyps reicht von 0 bis 2exp(32)-1. Die Einheit für den Datentyp TIME ist 1 ms.

Token
Das Netzwerk-"Token" steuert den vorübergehenden Besitz des Übertragungsrechtes durch einen einzelnen Teilnehmer. Das Token durchläuft die Teilnehmer in einer umlaufenden (aufsteigenden) Adressenfolge. Alle Teilnehmer verfolgen den Tokendurchlauf und können alle möglichen Daten erhalten, die mitgeschickt werden.

Traffic Cop
Der Traffic Cop ist ist eine Bestückungsliste, die aus der Benutzer-Bestückungsliste generiert wird. Der Traffic Cop wird in der SPS verwaltet und enhält zusätzlich zur Anwender-Bestückungsliste z.B. Status-Informationen zu den E/A-Stationen und Baugruppen.

Transition
Die Bedingung, bei der die Steuerung von einem oder mehreren Vorgängerschritten zu einem oder mehreren Nachfolgerschritten entlang einer gerichteten Verbindung übergeht.

Typisierte Literale

Wenn Sie den Datentyp für ein Literal selbst festlegen wollen, können Sie dies mit der folgenden Konstruktion tun: 'Datentypname'#'Wert des Literals'.

Beispiel
INT#15 (Datentyp: Integer, Wert: 15),
BYTE#00001111 (Datentyp: Byte, Wert: 00001111)
REAL#23.0 (Datentyp: Real, Wert: 23.0)

Für die Zuweisung des Datentyps REAL gibt es auch die Möglichkeit den Wert auf folgenden Weise anzugeben: 23.0.
Durch die Angabe der einer Kommastelle wird automatisch der Datentyp REAL zugewiesen.



UDEFB

Benutzer-definierte Elementare Funktionen/Funktionsbausteine
Funktionen oder Funktionsbausteine, die in der Programmiersprache C erstellt wurden, und die Concept in Bibliotheken zur Verfügung stellt.

UDINT

UDINT steht für den Datentyp "vorzeichenlose doppeltlange ganze Zahl (unsigned double integer)". Die Eingabe erfolgt als Integer-Literal, Basis-2-Literal, Basis-8-Literal oder Basis-16-Literal. Die Länge der Datenelemente ist 32 Bit. Der Wertebereich für Variablen dieses Datentyps reicht von 0 bis 2exp(32)-1.

UINT

UINT steht für den Datentyp "vorzeichenlose ganze Zahl (unsigned integer)". Die Eingabe erfolgt als Integer-Literal, Basis-2-Literal, Basis-8-Literal oder Basis-16-Literal. Die Länge der Datenelemente ist 16 Bit. Der Wertebereich für Variablen dieses Datentyps reicht von 0 bis (2exp 16)-1.

Unlocated Variable

Unlocated Variablen wird keine Signalspeicheradresse zugeordnet. Sie belegen somit auch keine Signalspeicheradressen. Der Wert dieser Variablen wird systemintern gespeichert und kann mit dem Referenzdaten-Editor gändert werden. Diese Variablen werden nur mit ihren symbolischen Namen angesprochen.

Signale die keinen Zugriff auf die Peripherie erfordern, z.B. Zwischenergebnisse, Systemmerker usw., sollten vorzugsweise als Unlocated Variable deklariert werden.



Variablen

Variablen dienen dem Datenaustausch innerhalb von Sections, zwischen mehreren Sections und zwischen dem Programm und der SPS.
Variablen bestehen mindestens aus einem Variablen-Namen und einem Datentyp. Wird einer Variablen eine direkte Adresse (Referenz) zugeordnet, spricht man von einer Located Variablen. Wird einer Variablen keine direkte Adresse zugeordnet, spricht man von einer Unlocated Variablen. Wird der Variablen ein Abgeleiteter Datentyp zugeordnet, spricht amn von einer Multielement-Variablen.
Außerdem gibt es noch Konstanten und Literale.

Verbindung

Eine Kontroll- oder Datenflußverbindung zwischen grafischen Objekten (z.B. Schritten im SFC-Editor, Funktionsbausteinen im FBD-Editor) innerhalb einer Section, grafisch als Linie dargestellt.



Warnung

Wird bei der Abarbeitung eines FFBs oder Schrittes ein kritischer Zustand erkannt (z.B. kritische Eingangswerte oder Zeitlimit überschritten), erfolgt eine Warnung, die Sie sich mit dem Menübefehl **Online** → **Ereignisanzeige...** ansehen können. Bei FFBs bleibt der ENO-Ausgang auf "1".

WORD

WORD steht für den Datentyp "Bit-Folge 16". Die Eingabe erfolgt als Basis-2-Literal, Basis-8-Literal oder Basis-16-Literal. Die Länge der Datenelemente ist 16 Bit. Ein numerischer Wertebereich kann diesem Datentyp nicht zugeordnet werden.



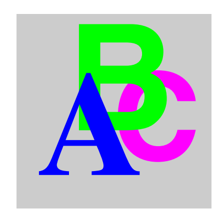
Zeitdauer-Literale

Erlaubte Einheiten für Zeitdauern (TIME) sind Tage (D), Stunden (H), Minuten (M), Sekunden (S) und Millisekunden (MS) oder Kombinationen hiervon. Die Zeitdauer muß durch das Präfix t#, T#, time# oder TIME# gekennzeichnet werden. Der "Überlauf" der höchstwertigen Einheit ist zulässig; z.B. ist die Eingabe T#25H15M erlaubt.

Beispiel
t#14MS, T#14.7S, time#18M, TIME#19.9H, t#20.4D, T#25H15M, time#5D14H12M18S3.5MS

Zwischenablage

Die Zwischenablage ist ein temporärer Speicher für ausgeschnittene oder kopierte Objekte. Diese Objekte können in Sections eingefügt werden. Bei jedem neuen Ausschneiden oder Kopieren wird der alte Inhalt der Zwischenablage überschrieben.



Index

I

M

O

P

R

S

T

U

Z